

Periodenübergreifende Budgetallokation in der Instandhaltung modularer  
Systeme – Ein Anwendungsfall von Deep Reinforcement Learning

**D i s s e r t a t i o n**

zur Erlangung des Doktorgrades

der Wirtschaftswissenschaften

vorgelegt von

Michael Krause

aus Dortmund

genehmigt von der Fakultät für Energie- und Wirtschaftswissenschaften

der Technischen Universität Clausthal,

Tag der mündlichen Prüfung

17. Juni 2021

Vorsitzender der Promotionskommission

Prof. Dr. Roland Menges

Hauptberichterstatter

Prof. Dr. Christoph Schwindt

Mitberichterstatter

Prof. Dr. Michael Kolonko

---

# Ehrenwörtliche Erklärung

Ich erkläre hiermit ehrenwörtlich, dass ich die vorliegende Arbeit selbstständig angefertigt habe. Die aus fremden Quellen direkt oder indirekt übernommenen Gedanken sind als solche kenntlich gemacht. Es wurden keine anderen als die angegebenen Quellen und Hinweise verwendet.

Die vorliegende Arbeit wurde bisher keiner anderen Prüfungsbehörde vorgelegt und auch noch nicht veröffentlicht.

Clausthal-Zellerfeld, der 25. April 2021

Michael Krause

---

## Danksagung

Ich möchte mich an dieser Stelle bei allen herzlich bedanken, die mich bei der Erstellung der vorliegenden Arbeit unterstützt haben. Mein Dank gilt insbesondere Herrn Professor Dr. Christoph Schwindt, der das Thema dieser Dissertation anregte und es mir ermöglichte, im Rahmen meiner Tätigkeit an seinem Lehrstuhl zu promovieren. Ich danke ihm dafür, dass er die wissenschaftliche Betreuung meiner Arbeit übernommen hat und diese durch kritische und inspirierende Fachdiskussionen bereicherte. Vielen Dank für die Begleitung während dieser langen Reise! Herrn Professor Dr. Michael Kolonko danke ich für die Übernahme des Korreferats und die wertvollen und hilfreichen Anregungen bei der Fertigstellung dieser Arbeit. Ein besonderer Dank gilt Prof. Dr. Lutz Angermann für die Hilfe bei der Bezwingung der Differentialgleichung. Für wertvolles Feedback möchte ich weiterhin Prof. Dr. Ehlers und Jun.-Prof. Dr. Marlin Ulmer danken. Ein besonderer Dank gilt Dr. Sven Arnold für die vielen guten Rückmeldungen und die aufbauenden Worte in der Endphase dieser Arbeit. Für stets gute Ratschläge und ein offenes Ohr für meine ganzen Dissertationsfragen danke ich Dr. Ingmar Schüle von ganzem Herzen. Schließlich danke ich meiner Familie – meinem Bruder (fürs Ehrgeiz wecken), meinen Kindern Svea und Nike (dass Ihr mir einen Grund gegeben habt, die Dissertation fertig zu stellen), meiner Mama (vielen Dank für die seelische und moralische Unterstützung), und vor allem meiner Ehefrau Nadine (für die Unterstützung, ohne die dieses Projekt nicht fertig gestellt worden wäre, und für Deine Liebe). Vielen Dank für die unendliche Geduld, die Ihr besonders in der letzten Zeit für mich aufgebracht habt.

---

## Kurzfassung

Eine funktionsfähige Infrastruktur ist zentrale Voraussetzung für die Wettbewerbsfähigkeit einer Volkswirtschaft. Daher sieht beispielsweise der Bundesverkehrswegeplan bis zum Jahr 2030 allein 141,6 Mrd. Euro für die Instandhaltung der bundesdeutschen Verkehrswege vor. Netzwerke von Straßen oder Brücken sind Beispiele für modulare Systeme, in denen der effiziente Ressourceneinsatz für Instandhaltungsmaßnahmen eine herausragende Rolle spielt. Hier möchte diese Arbeit einen Beitrag leisten, indem sie Planungsansätze vorstellt, die eine effiziente Budgetallokation auf einzelne Komponenten wie z.B. Straßenabschnitte und Perioden ermöglicht, um eine größtmögliche Leistungsfähigkeit des Gesamtsystems zu erreichen.

Die Festlegung einer geeigneten Budgetallokation für modulare Systeme unter Unsicherheit stellt eine anspruchsvolle Planungsaufgabe dar, die in der Literatur für unterschiedliche Annahmesysteme untersucht wurde. Besonderheiten der in der Dissertation untersuchten Problemstellung bestehen darin, dass die Planung sich auf einen mehrperiodigen Zeitraum bezieht, für den ein periodenübergreifendes Gesamtbudget zur Verfügung steht, und bei der Modellierung der zeitlichen Entwicklung der Komponentenzustände instationäre stochastische Verschleißprozesse zugelassen werden.

Die Wirkungen der Investitionen in Instandhaltungsmaßnahmen der Komponenten auf deren Zustände werden durch eine Funktion beschrieben, die sich als Lösung eines Anfangswertproblems für eine nichtlineare partielle Differentialgleichung ergibt. Die Abhängigkeit der Systemleistung von den Komponentenzuständen wird in Anlehnung an die in der Instandhaltungstheorie übliche Strukturfunktion über eine monoton wachsende Strukturbewertungsfunktion modelliert, wobei der Spezialfall einer quasikonkaven Strukturbewertungsfunktion vertiefend untersucht wird. Als Zielsetzung wird die Maximierung der minimalen Perioden-Systemleistung betrachtet.

Als erster Planungsansatz wird eine rollierende Planung untersucht, die unter Verwendung eines Erwartungswertmodells durchgeführt wird. Für die einperiodige Planung wird ein Grenznutzenalgorithmus entwickelt, der in einem Karush-Kuhn-Tucker-Punkt konvergiert, indem als homogener Grenznutzen positiver Investitionsentscheidungen der Lagrange-Multiplikator der Budgetrestriktion angenähert wird. Der Grenznutzen einer

---

Instandhaltungsinvestition ergibt sich dabei jeweils als Produkt aus aktueller Birnbaum-Importanz der Komponente und Effektivität der Investition. Für den Fall quasikonkaver Strukturbewertungsfunktionen konnte die globale Optimalität von Karush-Kuhn-Tucker-Punkten auch für den mehrperiodigen Fall bewiesen werden.

Der zweite Planungsansatz entspricht der Umsetzung einer flexiblen Planung zur Berechnung einer Politik, die für jede Periode und jedes Restbudget in Abhängigkeit eines beliebigen zulässigen Systemzustands eine gute Lösung für die zugehörige Budgetallokation liefert. Dazu wird das Optimierungsproblem als stochastisches dynamisches Programm aufgefasst, und die gesuchte Politik wird über die Gewichte eines tiefen neuronalen Netzes repräsentiert. Das Training dieses Netzes geschieht mittels Deep Reinforcement Learning, wobei das eingesetzte Actor-Critic-Verfahren „Deep Deterministic Policy Gradient“ spezifisch auf den Umgang mit überabzählbaren Zustands- und Aktionsräumen ausgelegt ist.

Die beiden Planungsansätze werden anhand von zufällig generierten Instanzen für verschiedene Systemstrukturen (Serien-, Parallel-, Serien-Parallel- und Brückenschaltungen) mit einer zufälligen Budgetallokation und untereinander verglichen. Aus diesem Vergleich werden Handlungsempfehlungen für die geeignete Budgetallokation abgeleitet. Mit den in der Arbeit entwickelten Methoden ist es erstmalig möglich, gute periodenübergreifende Budgetallokationen für Probleme der Instandhaltungsplanung modularer Systeme unter instationären Verschleißprozessen innerhalb realistischer Rechenzeiten von – je nach Planungsansatz – einigen Sekunden bis wenigen Stunden zu berechnen.

---

## Abstract

A functioning infrastructure is a key prerequisite for the competitiveness of an economy. For this reason, the German Federal Transport Infrastructure Plan, for example, earmarks 141.6 billion euros for the maintenance of Germany's transport routes alone by 2030. Networks of roads or bridges are examples of modular systems in which the efficient use of resources for maintenance measures plays a prominent role. This thesis aims to make a contribution by presenting planning approaches that allow efficient budget allocation to individual components such as road sections and periods in order to achieve the highest possible performance of the overall system.

Determining an appropriate budget allocation for modular systems under uncertainty is a challenging planning task that has been studied in the literature for different assumption systems. Special features of the problem studied in the dissertation are that the planning refers to a multi-period time frame for which a total budget spanning several periods is available, and non-stationary stochastic deterioration processes are allowed for when modeling the temporal evolution of the component states.

The effects of component maintenance investments on their states are described by a function that results as a solution to an initial value problem for a nonlinear partial differential equation. The dependence of the system performance on the component states is modeled by a monotonically increasing structure evaluation function following the structure function commonly used in maintenance theory, and the special case of a quasiconcave structure evaluation function is studied in depth. The maximization of the minimum period system performance is considered as the objective.

As a first planning approach, rolling planning is investigated using an expected value model. For single-period planning, a marginal utility algorithm is developed that converges to a Karush-Kuhn-Tucker point by approximating the Lagrange multiplier of the budget constraint as the homogeneous marginal utility of positive investment decisions. In each case, the marginal utility of a maintenance investment is the product of the current Birnbaum importance of the component and the effectiveness of the investment. For the case of quasiconcave structure valuation functions, the global optimality of Karush-Kuhn-Tucker points could also be proven for the multi-period case.

---

The second planning approach corresponds to the implementation of flexible planning to compute a policy that provides a good solution for the associated budget allocation for each period and residual budget depending on any feasible system state. For this purpose, the optimization problem is conceived as a stochastic dynamic program, and the policy sought is represented by the weights of a deep neural network. The training of this network is done using Deep Reinforcement Learning, and the Actor-Critic method "Deep Deterministic Policy Gradient" used is specifically designed to deal with overcountable state and action spaces.

The two planning approaches are compared using randomly generated instances for different system structures (series, parallel, series-parallel, and bridge networks) with a random budget allocation and with each other. From this comparison, recommended actions for appropriate budget allocation are derived. With the methods developed in the thesis, it is for the first time possible to compute good period-spanning budget allocations for problems of maintenance planning of modular systems under nonstationary deterioration processes within realistic computation times of - depending on the planning approach - a few seconds to a few hours.



# Inhaltsverzeichnis

<b>Vorspann</b>	<b>i</b>
Kurzfassung . . . . .	v
Abstract . . . . .	vii
Inhaltsverzeichnis . . . . .	ix
Symbolverzeichnis . . . . .	xiii
Abkürzungsverzeichnis . . . . .	xv
<b>1 Einleitung</b>	<b>1</b>
<b>2 Begriffliche Grundlagen und Planungsansätze für die Problemstellung</b>	<b>5</b>
2.1 Instandhaltung . . . . .	5
2.2 Budgetallokation in der Instandhaltung . . . . .	8
2.2.1 Definition . . . . .	8
2.2.2 Wirkung der Budgetallokation . . . . .	9
2.3 Modulare Systeme in der Instandhaltung . . . . .	11
2.4 Verschleiß in der Instandhaltung . . . . .	14
2.5 Entwicklung von Planungsansätzen . . . . .	17
2.5.1 Rollierende Planung . . . . .	19
2.5.2 Flexible Planung mit Hilfe neuronaler Netze . . . . .	20
<b>3 Forschungsstand</b>	<b>23</b>
3.1 Vorgehen bei der Recherche und Auswertung . . . . .	23
3.2 Überblick über den Stand der Forschung . . . . .	25
3.2.1 Budgetallokation bei monolithischen, mehrperiodigen Instandhaltungsmodellen . . . . .	25
3.2.2 Budgetallokation bei modularen, einperiodigen Instandhaltungsmodellen . . . . .	26

3.2.3	Budgetallokation bei modularen, mehrperiodigen Instandhaltungsmodellen . . . . .	28
3.2.4	Forschungslücke . . . . .	33
<b>4</b>	<b>Erwartungswertmodell für die rollierende Planung der Budgetallokation</b>	<b>35</b>
4.1	Mengen, Variablen und Parameter . . . . .	35
4.2	Verschleißmodellierung . . . . .	36
4.3	Modellierung der Dynamik, der Systemleistung und der Planungszielsetzung	38
4.3.1	Entwicklung der Komponentenzustände . . . . .	38
4.3.2	Strukturbewertung . . . . .	40
4.3.3	Planungszielsetzung . . . . .	42
4.4	Erwartungswertmodell . . . . .	43
<b>5</b>	<b>Analyse des Erwartungswertmodells</b>	<b>45</b>
5.1	Vorüberlegungen . . . . .	45
5.1.1	Volle Ausschöpfung des Budgets . . . . .	46
5.1.2	Herleitung einer geeigneten Zustandsübergangsfunktion . . . . .	46
5.1.3	Eigenschaften der Lösung der gewählten Zustandsübergangsfunktion	50
5.2	Analytische Lösung des Erwartungswertmodells für Zwei-Komponenten-Parallelsysteme bei einperiodigem Planungszeitraum . . . . .	53
5.2.1	Analyse für ausgewählte Verhältnisse der Skalierungsparameter der Investitionsfunktion . . . . .	56
5.2.2	Bewertung . . . . .	58
5.3	Ableitung einer Grenznutzenmethode für das einperiodige Erwartungswertmodell . . . . .	58
5.3.1	Notwendige Optimalitätsbedingungen erster Ordnung . . . . .	59
5.3.2	Grenznutzenmethode . . . . .	61
5.4	Analyse des mehrperiodigen Erwartungswertmodells . . . . .	63
5.4.1	Globale Optimalität quasikonkaver Zielfunktionen für einen Spezialfall . . . . .	64
5.4.2	Lösungsmethodik für allgemeine Strukturbewertungsfunktionen .	68

<b>6</b>	<b>Stochastisches dynamisches Optimierungsmodell</b>	<b>71</b>
6.1	Formulierung und Interpretation des stochastischen dynamischen Optimierungsproblems . . . . .	71
6.2	Ausgewählte Methoden des Reinforcement Learning . . . . .	75
6.2.1	Politikiteration, Wertiteration und Actor-Critic-Verfahren . . . . .	76
6.2.2	Überblick über wichtige Literatur zu Deep Reinforcement Learning	79
6.3	Umsetzung der flexiblen Planung mit Hilfe neuronaler Netze . . . . .	80
6.3.1	Vorüberlegungen . . . . .	80
6.3.2	Deep Reinforcement Learning: Deep Deterministic Policy Gradient (DDPG)-Algorithmus . . . . .	82
<b>7</b>	<b>Vergleichende experimentelle Performance-Analysen</b>	<b>89</b>
7.1	Konfiguration der Instanzen . . . . .	89
7.1.1	Strukturbewertungsfunktionen . . . . .	89
7.1.2	Zustandsübergangsfunktion . . . . .	93
7.1.3	Verschleißparametrisierung . . . . .	93
7.1.4	Anfangswertbedingungen . . . . .	94
7.2	Vergleiche für die starre Planung mit dem Erwartungswertmodell . . . . .	94
7.2.1	Vergleich periodenbezogener und periodenübergreifender Budgetallokation . . . . .	95
7.2.2	Weitere Analysen . . . . .	96
7.3	Vergleich verschiedener Planungsansätze . . . . .	100
7.3.1	Darstellung des Meta-Algorithmus . . . . .	100
7.3.2	Rechenergebnisse . . . . .	105
7.3.3	Regressionsanalyse . . . . .	107
7.3.4	Kritische Bewertung der Ergebnisse . . . . .	117
<b>8</b>	<b>Einordnung der Ergebnisse und offene Fragen</b>	<b>119</b>
8.1	Einordnung der Ergebnisse . . . . .	119
8.2	Methodenkritik . . . . .	120
8.2.1	Rollierende Planung . . . . .	120
8.2.2	Flexible Planung mit DDPG . . . . .	122
8.3	Implikationen und Empfehlungen . . . . .	123
8.4	Forschungsbedarf . . . . .	125

<b>9 Zusammenfassung</b>	<b>127</b>
<b>Literaturverzeichnis</b>	<b>130</b>
<b>A Mathematischer Anhang</b>	<b>146</b>
A.1 Verschleiß einfacher binärer Systeme . . . . .	146
A.2 Generalisierbarkeit der Investitionsfunktion . . . . .	148
A.3 Optimalitätsbedingungen . . . . .	149
A.3.1 Zwei-Komponenten-Parallelsystem, eine Periode . . . . .	149
A.3.2 Zwei-Komponenten-Seriensystem, eine Periode . . . . .	149
A.3.3 KKT-Bedingungen des einperiodigen Problems: Hinreichende Op- timalitätsbedingungen zweiter Ordnung . . . . .	154
A.3.4 Analyse der KKT-Bedingungen des mehrperiodigen Problems . .	156
A.4 Alternativbeweis: Minimum zweier strikt quasikonkaver Funktionen . .	159
A.5 Berechnung von Importanzen für die Brückenschaltung . . . . .	161
<b>B Experimentelle Performance-Analysen</b>	<b>162</b>
B.1 Übersichten der Instanzparameter . . . . .	162
B.2 Ausgewählter Quellcode . . . . .	168
B.2.1 Modellierung in GAMS . . . . .	168
B.2.2 Generierung der gerechneten Instanzen . . . . .	171
B.2.3 Einflussanalyse mit Lasso-Regression . . . . .	176
B.3 Reproduzierbarkeit . . . . .	180
B.4 Liste installierter Python-Pakete . . . . .	182
B.5 Weitere Ergebnis-Grafiken . . . . .	183
B.5.1 Fünfkomponentige Systeme . . . . .	183
B.5.2 Achtkomponentige Brückenschaltung . . . . .	200
B.6 Inputdaten für die Lasso-Regression . . . . .	205
B.7 Konstruierte Instanzanpassung . . . . .	207

## Symbolverzeichnis

$\gamma_i$	Grenznutzen der Komponente $i$
$\Gamma$	Gammafunktion
$\eta$	Effektivitätsfunktion
$\lambda$	Lagrangemultiplikator
$\pi$	Politik / Strategie
$\sigma^2$	Varianz
$\varphi$	Strukturfunktion
$\phi_t$	Systemleistung am Ende der Periode $t$
$\hat{a}$	Skalierungsparameter der Investitionsfunktion
$\hat{B}$	Zur Verfügung stehendes Gesamtbudget
$B_t$	Restbudget am Ende der Periode $t$
$\mathbb{E}$	Erwartungswert
$\exp$	Exponentialfunktion
$f$	Zielfunktion
$g_i$	Zustandsentwicklungsfunktion der Komponente $i$
$g_i^+$	Investitionsfunktion der Komponente $i$
$i$	Komponentenindex
$I_i$	Birnbaum-Importanz der Komponente $i$
$L$	Lagrangefunktion
$\lim$	Grenzwert (Limes)
$\ln$	natürlicher Logarithmus
$n$	Anzahl der Komponenten des betrachteten Systems
$\mathcal{O}$	Landausches Symbol
$\mathbb{P}$	Wahrscheinlichkeit
$\hat{p}_i$	Anfangszustand der Komponente $i$
$p_{it}$	Prozentualer Abnutzungsvorrat der Komponente $i$ am Ende der Periode $t$
$\mathbb{R}$	Menge aller reellen Zahlen
$\mathbb{R}_{\geq 0}$	Menge aller nichtnegativen reellen Zahlen
$\mathcal{R}$	Replay-Puffer
$R$	Spannweite
$\mathcal{S}$	Definitionsbereich der Zustandsvariable

$t$	Periodenindex
$T$	Länge des Planungshorizonts
$\mathcal{U}$	Gleichverteilung
$u$	Skalierungsparameter für Gamma-Verteilung
$U$	Nutzenfunktion
$v$	Formparameter für Gamma-Verteilung
$v(t)$	Formfunktion für Gamma-Prozess in Abhängigkeit der Zeit $t$
$\widehat{\Delta w}_{it}$	Erwarteter Verschleiss von Komponente $i$ innerhalb der Periode $t$
$\widetilde{W}$	Zufallsvariable für den Verschleiss
$\mathcal{X}$	Definitionsbereich der Entscheidungsvariable
$x_{it}$	Höhe der Instandhaltungsinvestition in Komponente $i$ zu Beginn der Periode $t$
$\frac{\partial}{\partial x_{it}}$	erste partielle Ableitung nach $x_{it}$
$\frac{\partial^2}{\partial x_{it}^2}$	zweite partielle Ableitung nach $x_{it}$
$\mathbf{0}$	Nullvektor
$\in$	ist Element der Menge
$\forall$	für alle
$\nabla$	Gradientenvektor
$\Rightarrow$	daraus folgt, dass
$\subset$	ist Teilmenge von oder gleich
$\supset$	ist Obermenge von oder gleich
$\top$	transponiert

Die Sortierung der Einträge erfolgte, soweit möglich, in alphabetischer Reihenfolge.

# Abkürzungsverzeichnis

Abb.	Abbildung
Alg.	Algorithmus
bspw.	beispielsweise
bzw.	beziehungsweise
CPU	Central Processing Unit
DDPG	Deep Deterministic Policy Gradient
engl.	englisch
f.	[und die] folgende [Dokumentseite]
ff.	[und] die folgenden [Dokumentseiten]
GAMS	General Algebraic Modeling System
GB	Giga Byte
GE	Geldeinheit
ggf.	gegebenenfalls
GHz	Giga Hertz
Gl.	Gleichung
GPU	Graphics Processing Unit
i. A.	im Allgemeinen

i. d. R.	in der Regel
inkl.	inklusive
Iter.	Iterationen
KKT	Karush-Kuhn-Tucker
km/h	Kilometer pro Stunde
LARS	Least Angle Regression Algorithmus
Lasso	Least absolut shrinkage and selection operator
lt.	laut
MARL	Multi-agent Reinforcement Learning
Max.	Maximiere
Min.	Minimiere
Mrd.	Milliarden
o. Ä.	oder Ähnliches
ReLU	Rectified Linear Unit
S.	Seite
s.	siehe
s. o.	siehe oben
sog.	sogenannte
Tab.	Tabelle
u. d. N.	unter den Nebenbedingungen
Ungl.	Ungleichung



usw.	und so weiter
vgl.	vergleiche
VS	Visual Studio
z. B.	zum Beispiel

# 1

## Kapitel 1

---

# Einleitung

Eine funktionsfähige Infrastruktur ist eine zentrale Voraussetzung für die Wettbewerbsfähigkeit einer Volkswirtschaft. Daher sieht beispielsweise der Bundesverkehrswegeplan bis zum Jahr 2030 allein 141,6 Mrd. Euro für die Instandhaltung der bundesdeutschen Verkehrswege vor (vgl. Bundesministerium für Verkehr und digitale Infrastruktur (BM-VI) (2020, S. 31)). In den USA beträgt der Investitionsstau für die Reparaturen von Straßen 435 Mrd. US-Dollar und 125 Mrd. US-Dollar für die Reparaturen von Brücken (vgl. American Society of Civil Engineers (ASCE) (2021)). Dies zeigt, dass der effiziente Ressourceneinsatz für Instandhaltungsmaßnahmen eine herausragende Rolle spielt. Netzwerke von Straßen oder Brücken sind Beispiele für modulare Systeme. Hier möchte diese Arbeit einen Beitrag leisten, indem sie Planungsansätze vorstellt, die eine effiziente Budgetallokation auf einzelne Komponenten wie z. B. Straßenabschnitte und Perioden ermöglicht, um eine größtmögliche Leistungsfähigkeit des Gesamtsystems zu erreichen.

Die Festlegung einer geeigneten Budgetallokation für modulare Systeme unter Unsicherheit (bezogen auf den nicht exakt vorherzusagenden Verschleiß der Komponenten des Systems) stellt eine anspruchsvolle Planungsaufgabe dar, die in der Literatur für unterschiedliche Annahmesysteme untersucht wurde. Die Besonderheiten der in der Dissertation untersuchten Problemstellung bestehen darin, dass die Planung sich auf einen mehrperiodigen Zeitraum bezieht, für den ein periodenübergreifendes Gesamtbudget zur Verfügung steht, und dass bei der Modellierung der zeitlichen Entwicklung der Komponentenzustände instationäre stochastische Verschleißprozesse zugelassen werden. Periodenübergreifend bedeutet, dass es keine festen Budgetbeschränkungen für die einzelnen Perioden des Planungszeitraums gibt.

Die Wirkungen der Investitionen in Instandhaltungsmaßnahmen auf die Zustände der einzelnen Komponenten werden durch eine Funktion beschrieben, die sich als Lösung eines Anfangswertproblems für eine nichtlineare partielle Differentialgleichung ergibt. Die Abhängigkeit der Systemleistung von den Komponentenzuständen wird in Anlehnung an

die in der Instandhaltungstheorie übliche Strukturfunktion über eine monoton wachsende Strukturbewertungsfunktion modelliert, wobei der Spezialfall einer quasikonkaven Strukturbewertungsfunktion vertiefend untersucht wird. Als Zielsetzung wird die Maximierung der minimalen Perioden-Systemleistung betrachtet.

Diese Dissertation stellt zwei Planungsansätze vor, die geeignet sind, auch bei zeitabhängigen Verschleißprozessen dieses Ziel bestmöglich zu erreichen.

Als erster Planungsansatz wird eine rollierende Planung untersucht, die unter Verwendung eines Erwartungswertmodells durchgeführt wird. Die notwendigen Optimalitätsbedingungen erster Ordnung gelten in Punkten, die den Karush-Kuhn-Tucker-(KKT-)Bedingungen genügen. Für die einperiodige Planung wird ein Grenznutzenalgorithmus entwickelt, der in einen solchen KKT-Punkt konvergiert, indem als homogener Grenznutzen positiver Investitionsentscheidungen der Lagrange-Multiplikator der Budgetrestriktion angenähert wird. Der Grenznutzen einer Instandhaltungsinvestition ergibt sich dabei jeweils als Produkt aus aktueller Birnbaum-Importanz der Komponente (dies ist die Ableitung der Strukturbewertungsfunktion) und Effektivität (dies ist die Ableitung der Zustandsübergangsfunktion) der Investition. Für den in dieser Arbeit angenommenen mehrperiodigen Fall soll zudem die globale Optimalität von KKT-Punkten für den Fall quasikonkaver Strukturbewertungsfunktionen bewiesen werden, um geeignete Solver für die Lösung des Planungsproblems begründet einsetzen zu können.

Der zweite Planungsansatz entspricht der Umsetzung einer flexiblen Planung zur Berechnung einer Politik, die für jede Periode und jedes Restbudget in Abhängigkeit eines beliebigen zulässigen Systemzustands eine gute Lösung für die zugehörige Budgetallokation liefert. Dazu wird das Optimierungsproblem als stochastisches dynamisches Programm aufgefasst, und die gesuchte Politik wird über die Gewichte eines tiefen neuronalen Netzes repräsentiert. Das Training dieses Netzes geschieht mittels Deep Reinforcement Learning, das diverse Ansätze zur Ermittlung solcher wie der hier gesuchten Politik bietet. Konkret eingesetzt wird in dieser Arbeit das Actor-Critic-Verfahren „Deep Deterministic Policy Gradient“, das spezifisch auf den Umgang mit überabzählbaren Zustands- und Aktionsräumen ausgelegt ist.

Die beiden Planungsansätze werden anhand von zufällig generierten Instanzen für verschiedene Systemstrukturen (Serien-, Parallel-, Serien-Parallel- und Brückenschaltungen) mit einer zufälligen Budgetallokation und untereinander verglichen. Aus den Ergebnissen dieses Vergleichs werden Handlungsempfehlungen für die geeignete Budgetallokation

abgeleitet. Mit den in der Arbeit entwickelten Methoden soll erstmalig ermöglicht werden, gute periodenübergreifende Budgetallokationen für Probleme der Instandhaltungsplanung modularer Systeme unter instationären Verschleißprozessen innerhalb realistischer Rechenzeiten von – je nach Planungsansatz – einigen Sekunden bis wenigen Stunden zu berechnen.

Um zu den angestrebten Erkenntnissen zu gelangen, ist die Arbeit wie folgt aufgebaut: Nach der Einleitung werden im zweiten Kapitel die theoretischen Grundlagen erläutert. Dazu gehören die Instandhaltung sowie in deren Kontext die Begriffe Budgetallokation, modulare Systeme sowie Verschleiß. Auf der Basis dieser Grundlagen wird das Planungsproblem formuliert und werden die im weiteren Lauf verwendeten Planungsansätze entwickelt.

Im dritten Kapitel wird der aktuelle Forschungsstand dargestellt. Dazu werden diverse Arbeiten zu Budgetallokation in der Instandhaltung kategorisiert und ausgewertet. Hieraus wird die bereits weiter oben als Besonderheit dieser Dissertation bezeichnete Konfiguration des Problems als Forschungslücke abgeleitet.

Im vierten Kapitel wird das Erwartungswertmodell als Grundlage der rollierenden Planung entwickelt. Dazu werden zuvor die dort verwendeten Mengen, Variablen und Parameter, die Modellierung des Verschleißes und die Modellierung der Dynamik, der Systemleistung und der Zielsetzung besprochen.

Das fünfte Kapitel analysiert das Erwartungswertmodell. Es werden zunächst allgemeine Eigenschaften bewiesen und eine geeignete Zustandsübergangsfunktion aus dem abnehmenden Grenznutzen finanzieller Mittel abgeleitet. Dann erweitert die Analyse Spezialfälle auf allgemeinere Strukturen mit einperiodigem Planungszeitraum und leitet so einen Grenznutzenalgorithmus ab. Zuletzt werden Voraussetzungen für die Optimalität des mehrperiodigen Erwartungswertmodells betrachtet.

Kapitel 6 formuliert das Planungsproblem als stochastisches dynamisches Optimierungsproblem und setzt so die Grundlage, um darauf aufbauende Methoden des Reinforcement Learning zu besprechen, die bei der Umsetzung der flexiblen Planung zum Einsatz kommen.

Kapitel 7 enthält die Rechenergebnisse für experimentelle Performance-Analysen auf insgesamt 25 zufallsgenerierten Instanzen. Für eine starre Planung wird anhand des Erwartungswertmodells gezeigt, wie viel Verbesserungspotenzial dadurch gegeben ist, eine

Budgetallokation nicht periodenbezogen, sondern periodenübergreifend durchzuführen. Im Anschluss werden die Planungsansätze mit einer zufälligen Budgetallokation und untereinander verglichen.

Im achten Kapitel werden die Ergebnisse eingeordnet und offene Fragen besprochen. Unter anderem findet dort eine Methodenkritik statt, und es werden Handlungsempfehlungen für potenzielle Nutzer abgeleitet. Am Ende des Kapitels werden offene Forschungsfragen formuliert.

Kapitel 9 bietet abschließend eine Zusammenfassung der gesamten Arbeit unter Darstellung der Ergebnisse der einzelnen Kapitel.

# 2 Begriffliche Grundlagen und Planungsansätze für die Problemstellung

In diesem Kapitel werden zunächst der für die vorliegende Arbeit grundlegende Begriff der Instandhaltung und in dessen Kontext Budgetallokation, modulare Systeme und Verschleiß näher erläutert. Im Bereich der Instandhaltung wird in Abschnitt 2.1 auf die Grundmaßnahmen und auf wichtige Instandhaltungsstrategien eingegangen. Abschnitt 2.2 beschäftigt sich mit der Aufgabe der Budgetallokation sowie den Grundlagen der Wirkung von Investitionen („abnehmender Grenznutzen finanzieller Mittel“). Im Abschnitt 2.3 werden modulare Systeme eingeführt. Gegenstand des Abschnitts 2.4 sind der Begriff des Verschleißes und Hinweise auf die Modellierung in der Instandhaltung. Mit diesen begrifflichen Grundlagen wird in Abschnitt 2.5 das Planungsproblem eingeführt, das in dieser Arbeit behandelt wird, und es werden Ansätze für dessen Lösung diskutiert.

## 2.1 Instandhaltung

*Instandhaltung* ist nach DIN-Normen-Ausschuss Dienstleistungen (NADL) (2019, S. 4) die „Kombination aller technischen und administrativen Maßnahmen sowie Maßnahmen des Managements während des Lebenszyklus [...] eines Objekts [...], die dem Erhalt oder der Wiederherstellung ihres funktionsfähigen Zustands dient, sodass es die geforderte Funktion [...] erfüllen kann“. Da nach DIN-Normen-Ausschuss Dienstleistungen (NADL) (2019, S. 12) die Instandhaltung in die Grundmaßnahmen Wartung, Inspektion, Instandsetzung und Verbesserung unterteilt werden kann, seien diese Begriffe im Folgenden ebenfalls definiert.

*Wartung* bezeichnet laut DIN-Normen-Ausschuss Dienstleistungen (NADL) (2019, S. 5) „Maßnahmen zur Verzögerung des Abbaus des vorhandenen Abnutzungsvorrats“. In dieser Arbeit bezeichnen wir den Abbau des Abnutzungsvorrats als *Verschleiß* (zum Verschleißbegriff vgl. Abschnitt 2.4).

*Inspektion* ist die „Prüfung auf Konformität der maßgeblichen Merkmale eines Objekts [...], durch Messung, Beobachtung oder Funktionsprüfung“ [DIN-Normen-Ausschuss Dienstleistungen (NADL) (2019, S. 5)]. Dadurch wird der Zustand des jeweiligen Objekts eingeschätzt. Ein Objekt kann dabei ein Teil, ein Bauelement o. Ä. sein. In der vorliegenden Arbeit bezeichnen wir ein solches Objekt mit dem Begriff *Komponente* (vgl. Abschnitt 2.3 zu modularen Systemen).

*Instandsetzung* bezeichnet eine „physische Maßnahme, die ausgeführt wird, um die Funktion eines fehlerhaften Objekts [...] wiederherzustellen“ [DIN-Normen-Ausschuss Dienstleistungen (NADL) (2019, S. 6)]. Dies bezeichnet meist eine Reparatur, es kann sich aber auch um eine fehlerbedingte Ersetzung des Objekts handeln.

*Verbesserung* hingegen ist die „Kombination aller technischen und administrativen Maßnahmen sowie Maßnahmen des Managements zur Steigerung der immanenten Zuverlässigkeit und/oder Instandhaltbarkeit und/oder Sicherheit eines Objekts [...]“ [DIN-Normen-Ausschuss Dienstleistungen (NADL) (2019, S. 6 f.)]. Diese Grundmaßnahme werden wir nicht weiter betrachten.

Neben den Grundmaßnahmen nach DIN-Normen-Ausschuss Dienstleistungen (NADL) (2019, S. 12) ist in der Instandhaltungstheorie noch die *vorbeugende Ersetzung* (*Erneuerung*) besonders bedeutsam, die die Vermeidung von Ausfällen eines Systems durch die Ersetzung intakter Komponenten beschreibt.

Da diese Arbeit den Fokus nicht auf die Funktionsfähigkeit von Komponenten im Sinne der Definition von Instandsetzung legt, sondern auf die Erhöhung des Abnutzungsvorrats (und somit auf die Verzögerung des Abbaus), wird in dieser Arbeit der Instandhaltungsbegriff vor allem als Wartung aufgefasst. Wir gehen davon aus, dass wir zu jedem Zeitpunkt den Zustand der einzelnen Komponenten bezüglich des verbleibenden Abnutzungsvorrats kennen. Dies kann beispielsweise über geeignete Sensorik erfolgen; andernfalls nehmen wir an, dass realistische Schätzungen über den Abnutzungsvorrat zu jedem betrachteten Zeitpunkt und zu jeder Komponente vorliegen.

Als *Instandhaltungsstrategie* wird die Vorgehensweise des Managements bezeichnet, die angewendet wird, um die Instandhaltungsziele zu erreichen (vgl. DIN-Normenausschuss Dienstleistungen (NADL) (2018, S. 9)). Dies kann mittels verschiedener *Instandhaltungsarten* geschehen. In der Literatur finden sich alternative Kategorisierungen von Instandhaltungsarten (zum Beispiel in Ryll und Freund (2010, S. 30 f.) oder in Leidinger (2017, S. 16 ff.)), die sich zum Teil widersprechen. In dieser Arbeit halten wir uns an die Kategorisierung der deutschen bzw. europäischen Norm. Hier wird u. a. zwischen präventiver (engl. preventive), vorausbestimmter (engl. predetermined), zustandsorientierter (engl. condition-based) und voraussagender (engl. predictive) Instandhaltung unterschieden:

*Präventive Instandhaltung* ist dort definiert als eine „Instandhaltung zur Beurteilung und / oder Verminderung von Abbau und zur Reduzierung der Ausfallwahrscheinlichkeit eines Objekts“ [DIN-Normenausschuss Dienstleistungen (NADL) (2018, S. 34)].

*Vorausbestimmte Instandhaltung* ist eine „präventive Instandhaltung, durchgeführt in festgelegten Zeitabständen oder nach einer festgelegten Zahl von Nutzungseinheiten“ [DIN-Normenausschuss Dienstleistungen (NADL) (2018, S. 35)].

*Zustandsorientierte Instandhaltung* beschreibt hingegen „präventive Instandhaltung, die die Beurteilung des physischen Zustands, Analysen und die möglichen, daraus resultierenden Instandhaltungsmaßnahmen beinhaltet“ [DIN-Normenausschuss Dienstleistungen (NADL) (2018, S. 35)].

*Voraussagende Instandhaltung* bezeichnet eine „zustandsorientierte Instandhaltung, die nach einer Vorhersage, abgeleitet von wiederholter Analyse oder bekannten Eigenschaften und Bestimmung von wichtigen Parametern, welche den Abbau des Objekts kennzeichnen, durchgeführt wird“ [DIN-Normenausschuss Dienstleistungen (NADL) (2018, S. 35)].

Ab Kapitel 4 dieser Arbeit werden wir ein Modell aus der Klasse der voraussagenden Instandhaltung entwickeln. Zum einen gehen wir davon aus, den Zustand zu jedem Zeitpunkt zu kennen. Zum anderen leiten wir die Maßnahmen aus bekannten Parametern (insbesondere dem Erwartungswert) und der Simulation (also wiederholter Analyse) stochastischer Verteilungen ab.



## 2.2 Budgetallokation in der Instandhaltung

Zunächst definieren wir den Begriff Budgetallokation und werfen einen Blick auf die Wirkung von Investitionen im Rahmen der Budgetallokation. Dabei setzen wir unsere Betrachtungen in den Kontext der Instandhaltung.

### 2.2.1 Definition

Definiert werden *Budgets* in Koschnick (1996, S. 88 f.) als „schriftliche Zusammenfassungen, durch welche den Aufgabenträgern, insbesondere dem Management in einer Organisation [...] für einen abgegrenzten Zeitraum fixierte Sollgrößen im Sinne von Soll-Ergebnissen geplanter Aktivitäten in wertmäßiger und mitunter auch mengenmäßiger Form vorgegeben werden sollen“. In dieser Arbeit verwenden wir die wertmäßige Form, d. h. eine bestimmte Menge an finanziellen Mitteln wird für die Durchführung von Instandhaltungsmaßnahmen zur Verfügung gestellt.

Nach Koschnick (1996, S. 89) ist *Budgetallokation* „die Allokation eines Budgets auf einzelne Positionen nach sachlichen und zeitlichen Gesichtspunkten.“ In unserem Kontext von modularen Systemen in der Instandhaltung (vgl. Abschnitt 2.3) ist somit zu bestimmen, wie ein vorgegebenes Budget auf die Wartung für einzelne Komponenten des modularen Systems (sachliche Allokation) und in einzelnen Perioden eines Planungszeitraums (zeitliche Allokation) verteilt werden sollte.

Im Gegensatz dazu umfasst *Budgetierung* „alle Aufgaben, die die Erstellung, Verabschiedung und Kontrolle von Budgets betreffen. Ergebnis der Budgetierung ist die wertmäßige Zusammenfassung der geplanten Entwicklung der Unternehmung in einer zukünftigen Geschäftsperiode“ [Koschnick (1996, S. 89)]. In diesem Sinne ist Budgetierung nicht Bestandteil dieser Arbeit.

In der Literatur findet man häufig Beispiele, bei denen ein Budgetplan ein Budget auch schon für einzelne Perioden eines mehrperiodigen Planungszeitraums fest vorgibt, z. B. für die jeweiligen Jahre eines Fünfjahresplans. Durch diese Art der Festlegung geht allerdings Optimierungspotenzial verloren, ein Umstand, den wir später noch einmal in Abschnitt 7.2.1 aufgreifen werden.

Abbildung 2.1 illustriert das vorliegende Problem der Budgetallokation. Ein Gesamtbudget wird dabei beispielhaft auf zwei Komponenten in zwei Perioden für die Instandhaltung

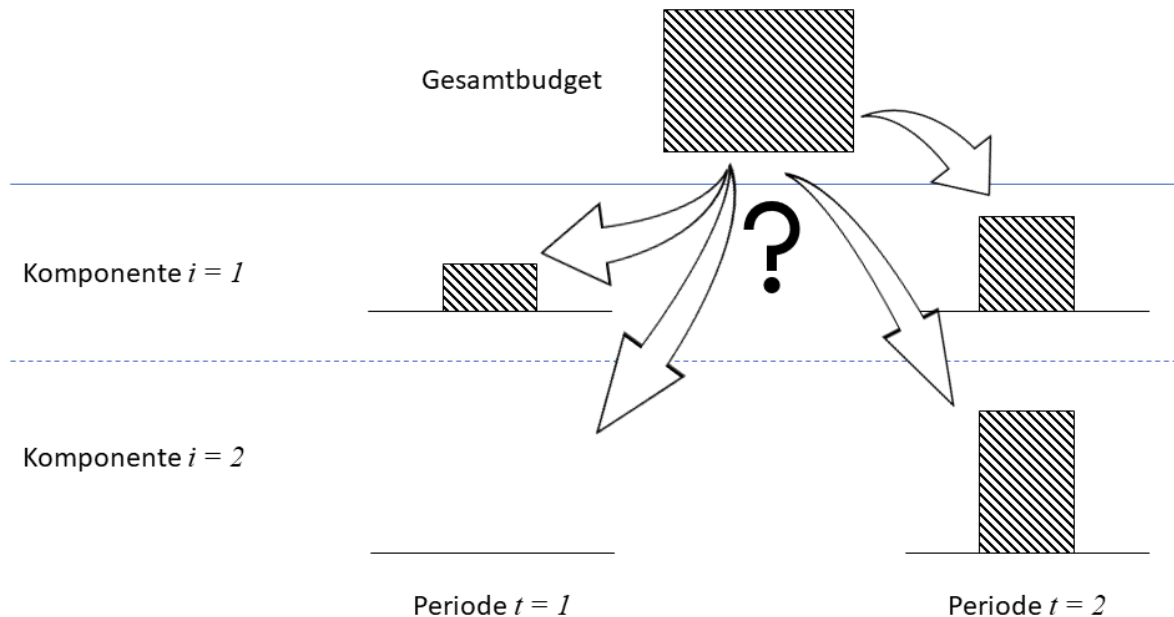


Abb. 2.1: Beispielhafte Budgetallokation eines Gesamtbudgets auf zwei Komponenten in zwei Perioden

Quelle: Eigene Darstellung

dieser Komponenten aufgeteilt, wobei es auch möglich ist, dass nicht jeder Komponenten-Perioden-Kombination ein positiver Anteil zugewiesen wird (im Beispiel Komponente  $i = 2$  in Periode  $t = 1$ ). Das Fragezeichen zeigt an, dass die optimale Aufteilung auf die Komponenten und Perioden zu bestimmen ist. Dieses Problem ist nicht trivial, da für eine optimale Aufteilung u. a. auch die Wirkung des eingesetzten Budgets zu berücksichtigen ist.

### 2.2.2 Wirkung der Budgetallokation

Nach dem ersten Gossenschen Gesetz („Gesetz vom abnehmenden Grenznutzen“, vgl. van Suntum (2005, S. 35 ff.)) nimmt der Nutzen von jeder zusätzlichen Einheit, die man aufwendet, stets ab. Das bedeutet, dass der Nutzen der ersten Einheit im Allgemeinen größer ist als der der zweiten Einheit, der Nutzen der zweiten Einheit größer als der der dritten usw.

Das folgende Beispiel soll das Gesetz im Kontext von Instandhaltung veranschaulichen. Es

sei angenommen, man habe Budget zur Verfügung, um den Straßenbelag eines Abschnitts einer Bundesstraße auszubessern. Das aktuelle Tempolimit liege aufgrund des schlechten Straßenzustands bei 30 km/h. Die erste Geldeinheit (z. B. 10.000 Euro) würde dann dazu verwendet, die Schlaglöcher auszubessern, wodurch auf diesem Straßenabschnitt wieder 60 km/h gefahren werden dürften. Die zweite Geldeinheit würde zur flächigen Neuteerung des Straßenabschnitts verwendet, wodurch der Straßenabschnitt wieder mit 80 km/h befahrbar wäre. Wir bilden nun den Nutzen über kardinale Werte<sup>1</sup>, im vorliegenden Fall die Werte der Geschwindigkeitsbegrenzungen, ab. Der Nutzenzuwachs (in km/h) wäre also für die erste Einheit 30, für die zweite nur 20. Im weiteren Verlauf dieser Arbeit soll der prozentuale Abnutzungsvorrat den Nutzen einer Komponente darstellen. Wenn wir im vorangehenden Beispiel bleiben und eine maximale Geschwindigkeit von 100 km/h auf dem Straßenabschnitt annehmen, die gleichbedeutend mit einem neuwertigen Zustand wäre, dann hätten wir entsprechend zunächst einen Nutzenzuwachs von 0,3 (also 30 %) auf 0,6 und dann von 0,6 auf 0,8.

Die Annahme, dass das Gossensche Gesetz auch in Bezug auf Instandhaltungsmaßnahmen gilt, ist für die spätere Modellierung in dieser Arbeit von entscheidender Bedeutung, da der maximale Abnutzungsvorrat in unserem Modell bei 100 % (gleichbedeutend mit einem neuwertigen Zustand) einer Komponente liegt und wir davon ausgehen, dass ein beliebig hoher Geldeinsatz höchstens den neuwertigen Zustand erreichen kann. Würden wir eine (positiv) proportionale Wirkung des Geldes unterstellen, so würde man Werte über 100 % erreichen können. Insbesondere liegt eine überabzählbar große Aktionsmenge vor, da prinzipiell jede positive Investition in die Instandhaltung von Komponenten möglich ist.

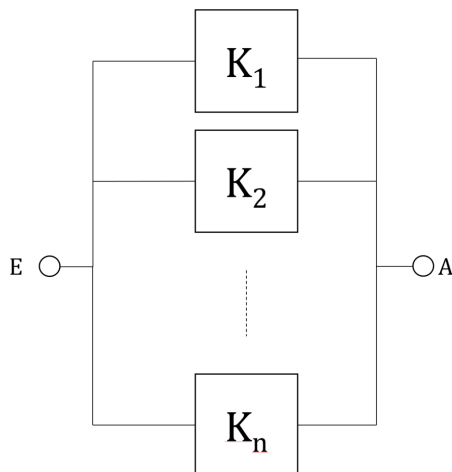
Neben den Betrachtungen für nur eine Komponente (wie etwa dem eben erwähnten Straßenabschnitt) wird in der vorliegenden Dissertation die Wirkung des Geldeinsatzes auf das Zusammenspiel von mehreren Komponenten eines Systems (s. Abschnitt 2.3) unter Berücksichtigung von Verschleiß (s. Abschnitt 2.4) im Zeitverlauf berücksichtigt. Dies werden wir nun weiter beleuchten.

---

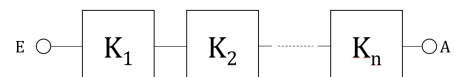
<sup>1</sup>Kardinale Werte sind Werte auf einer Kardinalskala bzw. metrischen Skala, mit denen sich sowohl Reihenfolgen also auch quantifizierbare Abstände bilden lassen.

## 2.3 Modulare Systeme in der Instandhaltung

Ein *System* ist nach Feess und Gillenkirch (2018) die „Menge von geordneten Elementen mit Eigenschaften, die durch Relationen verknüpft sind. Die Menge der Relationen zwischen den Elementen eines Systems ist seine Struktur.“ Weiter definieren Feess und Gillenkirch (2018) ein *Element* als „Bestandteil eines Systems, der innerhalb dieser Gesamtheit nicht weiter zerlegt werden kann.“ In dieser Arbeit verwenden wir die Begriffe *Elemente* und *Komponenten* synonym. Weiter sagen wir, ein System sei dann *modular*, wenn es mindestens zwei Komponenten besitzt, für die individuelle Entscheidungen getroffen werden können. Wir gehen also davon aus, dass alle Komponenten individuell instandgehalten werden können. Wenn sich Entscheidungen auf alle vorhandenen Komponenten gleich auswirken (oder es nur eine Komponente gibt), so bezeichnen wir das System als *monolithisch*. Ein modulares System ist also mehr als die Summe seiner Teile; es kommt bei einem modularen System insbesondere auf das Zusammenwirken der Komponenten an.



(a) Parallelsystem



(b) Seriensystem

Abb. 2.2: Zuverlässigkeitsschaltbilder von Parallel- und Seriensystem

Quelle: Eigene Darstellung in Anlehnung an Beichelt (1993, S. 45 ff.)

Eine übliche grafische Darstellung von modularen Systemen im Bereich der Instandhaltung sind *Zuverlässigkeitsschaltbilder*<sup>2</sup> (s. Abb. 2.2). Ein Zuverlässigkeitsschaltbild ist ein Graph mit einer Quelle  $E$  und einer Senke  $A$ . Die Knoten  $E$  und  $A$  werden auch Eingangs-

<sup>2</sup>*Zuverlässigkeit* fassen wir als „Intaktwahrscheinlichkeit“ auf, konkret als die Wahrscheinlichkeit, dass eine Komponente oder ein System zu einem bestimmten Zeitpunkt noch nicht defekt ist.

und Ausgangsknoten genannt. Jeder weitere Knoten des Graphen verkörpert genau ein Element des Systems, und jedes Element wird durch mindestens einen Knoten verkörpert. Ein Graph dieser Art heißt Zuverlässigkeitsschaltbild des betrachteten Systems, wenn er folgende Eigenschaft besitzt: Das System ist genau dann funktionstüchtig, wenn mindestens ein Weg von  $E$  nach  $A$  führt, der nur solche weiteren Knoten enthält, die funktionstüchtigen Elementen entsprechen (vgl. Beichelt (1993, S. 45)). So stellt Abb. 2.2a ein Parallelsystem dar, in dem nur einer der inneren Knoten ein funktionstüchtiges Element verkörpern muss, damit das System funktioniert. Daneben stellt Abb. 2.2b ein Seriensystem dar, in dem alle inneren Knoten funktionstüchtige Elemente verkörpern müssen, damit ein intakter Weg von  $E$  nach  $A$  existiert und das System funktioniert.

Das Konzept der Zuverlässigkeit bezieht sich in der Instandhaltung i. d. R. auf binäre Systeme. In einem binären System können die Komponenten ausschließlich die beiden Zustände „funktionstüchtig“ und „defekt“ annehmen.<sup>3</sup> Würde man die Zuverlässigkeiten der Komponenten dieser binären Systeme als Zustände interpretieren, so würde aus einem binären System ein Multizustandssystem. Multizustandssysteme sind nach Lisnianski u. a. (2010, S. 8 ff.) also Systeme, deren Komponenten mehr als zwei verschiedene Zustände annehmen können.

Wir abstrahieren in dieser Arbeit von dem Konzept der Zuverlässigkeit und nehmen stattdessen den prozentualen Abnutzungsvorrat der Komponenten als deren Zustand an. Der Grund dafür liegt vor allem in der Wahl der Verschleißmodellierung (vgl. Abschnitt 2.4). Das bedeutet, dass die betrachteten Systeme in dieser Arbeit zur Klasse der Multizustandssysteme gehören, und lässt uns die o. g. Darstellungsform der Zuverlässigkeitsschaltbilder für die untersuchten Systeme im weiteren Verlauf verwenden. Weiter wird wie bei Zuverlässigkeiten angenommen, dass jede Komponente überabzählbar viele verschiedene Zustände annehmen kann, konkret jeden Wert zwischen 0 % und 100 %.

Anwendungsbeispiele für die Instandhaltung modularer Systeme gibt es viele. Die ersten größeren Entscheidungsunterstützungssysteme der Instandhaltung wurden bspw. für die Unterhaltung von Straßennetzwerken entwickelt (vgl. Golabi u. a. (1982)) oder für die Instandhaltung eines Netzes von Brücken in den USA (vgl. Golabi und Shepard (1997)). Hier werden die Straßen bzw. Brücken als Komponenten des Systems interpretiert.

Zwischen den Komponenten eines Systems können aus Sicht der Instandhaltung vielfältige

---

<sup>3</sup>Zu Verschleißbegriffen von binären Systemen mit nur einer Komponente siehe Anhang A.1.

Abhängigkeiten bestehen. Olde Keizer u. a. (2017) unterscheiden bei Instandhaltung von modularen Systemen zwischen vier Typen von Abhängigkeiten, die jeweils weiter untergliedert werden (vgl. Abb. 2.3 auf Seite 15).

*Strukturelle Abhängigkeiten* betreffen die langfristigen, statischen Beziehungen zwischen verschiedenen Komponenten. Sie untergliedern sich in technische Abhängigkeiten und Performance-Abhängigkeiten. Technische Abhängigkeiten bedeuten, dass bestimmte technische Konfigurationen zu Beschränkungen der Instandhaltung oder Nutzung führen können. Zum einen kann die Instandhaltung einer Komponente dazu führen, dass eine andere Komponente ebenfalls instandgehalten werden muss oder nicht instand gehalten werden darf. Zum anderen ist es möglich, dass die Instandhaltung einer Komponente sich auf den Betrieb anderer Komponenten auswirkt, z. B. bei einer Maschine, die dann gestoppt werden muss. Performance-Abhängigkeiten sind Abhängigkeiten von Komponenten über die Leistung des Gesamtsystems. So betrifft in einem Seriensystem der Ausfall einer Komponente das gesamte System, in einem Parallelsystem hingegen nicht. Für die vorliegende Dissertation ist die letztere Form struktureller Abhängigkeiten von Bedeutung, also die der Performance-Abhängigkeiten.

*Stochastische Abhängigkeiten* bezeichnen die Tatsache, dass Ausfall- bzw. Verschleißprozesse von Komponenten voneinander abhängen können. Stochastische Abhängigkeiten lassen sich unterteilen in ausfallbedingte Schäden, Lastteilung und Gleichtakt-Verschleiß. Abhängigkeiten aufgrund ausfallbedingter Schäden treten bei Ausfall einer Komponente auf. Dieser Ausfall kann einen Schaden an anderen Komponenten verursachen, was zu einer Erhöhung des Verschleißgrads oder sogar zu einem sofortigen Ausfall jener Komponenten führen kann. Abhängigkeiten aufgrund von Lastteilung treten auf, wenn sich mehrere Komponenten die Systemlast teilen. Wenn eine Komponente ausfällt, arbeitet das System weiter, aber die übrigen Komponenten werden bei gleicher Systemleistung stärker beansprucht. Der Ausfall einer Komponente erhöht somit die Belastung der verbleibenden funktionierenden Komponenten, die dadurch schneller verschleifen. Abhängigkeiten aufgrund von Gleichtakt-Verschleiß bedeuten, dass mehrere Komponenten ähnlichen Arbeitsbedingungen unterliegen. Sie können gleichzeitig ausfallen oder ähnlich verschleifen. Eine Zunahme des Verschleißgrads für eine Komponente geht oft mit einer Zunahme des Verschleißgrads anderer Komponenten einher. Stochastische Abhängigkeiten bestehen bei dem in der vorliegenden Dissertation behandelten Problem nicht, da ein unabhängiger Verschleiß der einzelnen Komponenten angenommen wird.

*Ressourcenabhängigkeiten* treten dann auf, wenn sich Komponenten gemeinsame Ressourcen teilen. Dabei können Abhängigkeiten in den Bereichen Personal, Werkzeuge, Ersatzteile, Transport oder Budget auftreten. Üblicherweise haben (Sub-)Unternehmen Instandhaltungsabteilungen oder -teams, die für die Instandhaltung verantwortlich sind. Da für jede Instandhaltungsmaßnahme i. d. R. mindestens eine Person erforderlich ist, ist die Anzahl der gleichzeitig durchführbaren Instandhaltungsaktivitäten durch die Anzahl der verfügbaren Arbeitskräfte begrenzt. Ähnlich verhält es sich, wenn eine begrenzte Anzahl von Werkzeugen vorhanden ist. Dann können ebenfalls häufig nicht alle Instandhaltungsaktivitäten simultan durchgeführt werden. Beispielsweise muss für ein einzelnes Werkzeug, wenn es für mehrere Arbeiten benötigt wird, eine Entscheidung über die Reihenfolge getroffen werden, in der die Komponenten instandgehalten werden. Wenn mehrere Komponenten sich Ersatzteile teilen, die nicht in ausreichender Anzahl vorhanden sind, muss eine Priorisierung durchgeführt werden. Eine erhöhte Schwierigkeit stellen bei Ressourcenabhängigkeiten die zusätzliche Einplanung von Liefer- bzw. Bereitstellungszeiten dar.

Eine für diese Arbeit relevante Ressourcenabhängigkeit ergibt sich offensichtlich durch die Begrenzung des Budgets. Instandhaltungsaktivitäten verursachen Kosten und müssen sich i. d. R. ein gemeinsames Budget teilen, wobei die Länge der Budgetperioden in der Praxis unterschiedlich ist.

*Ökonomische Abhängigkeiten* bedeuten, dass die kombinierte Instandhaltung mehrerer Komponenten entweder teurer (negative ökonomische Abhängigkeit) oder kostengünstiger (positive ökonomische Abhängigkeit) ist als die Instandhaltung jeder Komponente einzeln. Diese Abhängigkeit wird in der vorliegenden Dissertation ausgeblendet.

Das in dieser Arbeit behandelte Problem weist somit für die einzelnen Komponenten sowohl strukturelle Performance- als auch Ressourcenabhängigkeiten bezüglich des Budgets auf.

## 2.4 Verschleiß in der Instandhaltung

*Verschleiß* ist der fortschreitende Materialverlust aus der Oberfläche eines festen Körpers, hervorgerufen durch mechanische Ursachen, d. h. Kontakt und Relativbewegung eines festen, flüssigen oder gasförmigen Gegenkörpers (vgl. Normenausschuß Materialprüfung

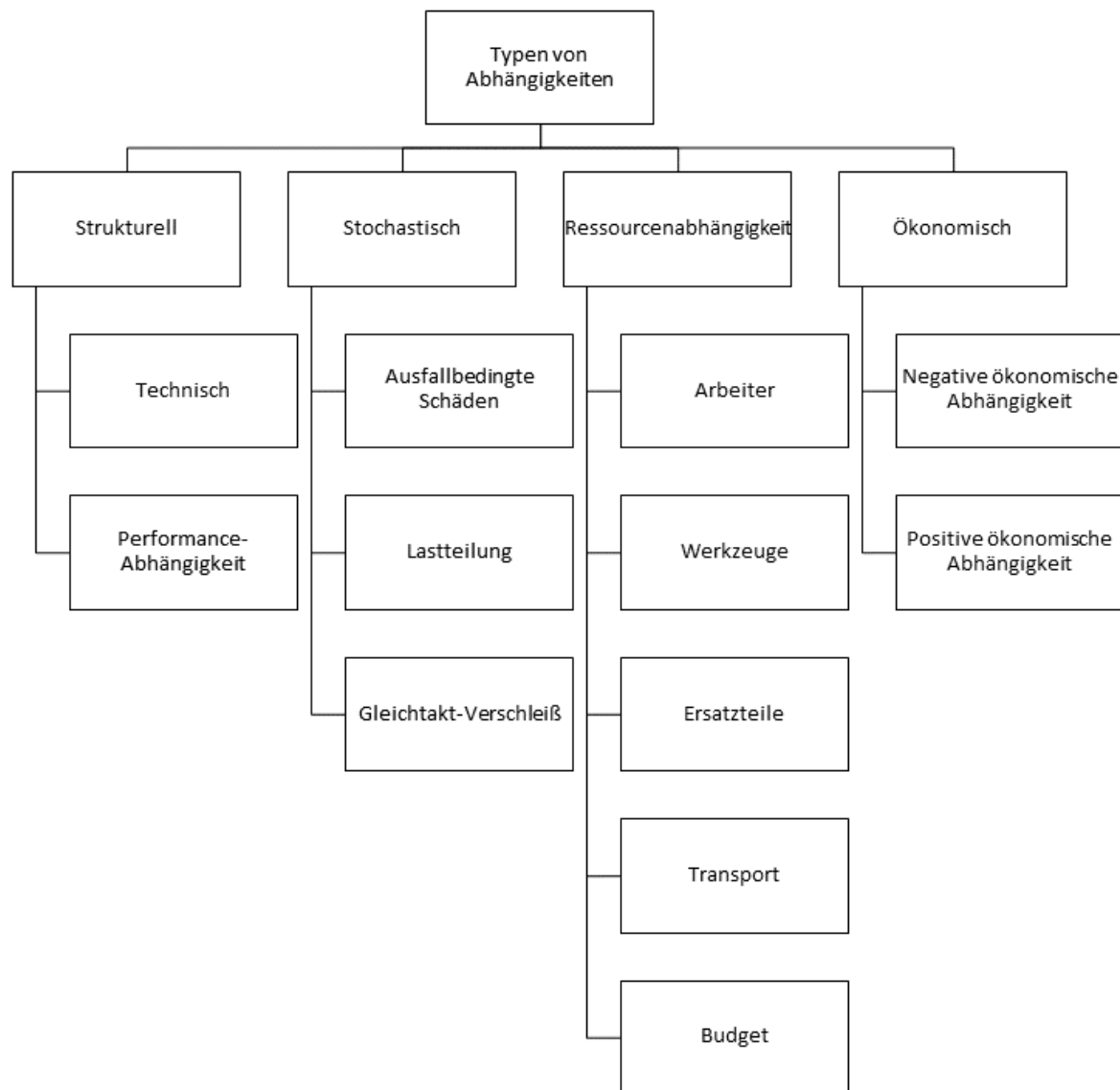


Abb. 2.3: Typen von Abhängigkeiten in modularen Systemen

Quelle: Eigene Darstellung in Anlehnung an Olde Keizer u. a. (2017)



(NMP) im DIN Deutsches Institut für Normung e.V. (1979)). Der Begriff wird im Kontext von Instandhaltung häufig synonym mit dem Begriff *Abnutzung* verwendet, der laut DIN-Normen-Ausschuss Dienstleistungen (NADL) (2019, S. 7) den „Abbau des Abnutzungsvorrates [...], hervorgerufen durch chemische und/oder physikalische Vorgänge“ beschreibt. In DIN-Normen-Ausschuss Dienstleistungen (NADL) (2019, S. 8) ist Abnutzungsvorrat definiert als „Vorrat der möglichen Funktionserfüllungen unter festgelegten Bedingungen, der eines Objekts [...] aufgrund der Herstellung, Instandsetzung [...] oder Verbesserung [...] innewohnt“.

In dieser Arbeit betrachten wir einen graduellen (physischen) Verschleiß, der bspw. über Sensorik erfasst werden kann. Daher wird in dieser Arbeit der (prozentuale) Abnutzungsvorrat mit dem Zustand einer Komponente assoziiert. Das bedeutet insbesondere, dass ein Zustand einer Komponente kardinal bewertet und ins Verhältnis zum neuwertigen Zustand gesetzt wird, welcher bei 100 % liegt.

Da Verschleiß i. d. R. nicht vollkommen vorhersagbar ist, legt man für die Modellierung einen stochastischen Prozess zugrunde. Wir nehmen in dieser Arbeit einen graduellen Verschleiß an, der durch kontinuierliche Nutzung entsteht. Für diese Art von Verschleiß wird in der Literatur der Gamma-Prozess als geeignet angesehen (vgl. van Noortwijk (2009)). In der Instandhaltung ist für diesen Prozess der Erwartungswert des kumulierten Verschleißes im Zeitintervall  $[0; t]$  laut van Noortwijk (2009) häufig proportional zu einer Funktion  $a \cdot t^b$  in Abhängigkeit der Zeit  $t$  mit Konstanten  $a > 0$  und  $b > 0$ .

In der Praxis liegen insbesondere für den Exponenten  $b$  in vielen Fällen schon Erfahrungswerte vor. So ist beispielsweise der erwartete Verschleiß von Beton durch Korrosion für eine Armierung linear (also  $b = 1$ ), für einen Säureangriff auf Beton parabolisch mit  $b = 2$  und bei diffusionskontrollierter Alterung des Betons über die Quadratwurzel mit  $b = 0,5$  abbildbar (vgl. van Noortwijk (2009); Ellingwood und Mori (1993)). Gemäß dieser Modellierung ist also sowohl der erwartete kumulierte Verschleiß als auch der Erwartungswert für die Differenz zweier aufeinanderfolgender Perioden<sup>4</sup> für  $b \neq 1$  abhängig von der Periode  $t$ . Das bedeutet insbesondere, dass wir in der Instandhaltung auch Verschleißprozesse annehmen müssen, deren Verteilungen nicht zeitunabhängig und damit nicht stationär sind.

---

<sup>4</sup>Für die Perioden  $t$  und  $t + 1$  wäre das z. B.  $a \cdot (t + 1)^b - at^b$ .

## 2.5 Entwicklung von Planungsansätzen

Nach Scholl (2001, S. 9) ist Planung „ein von Entscheidungsträgern auf der Grundlage unvollkommener Informationen durchgeführter, zukunftsorientierter, grundsätzlich systematischer und rationaler Prozess zur Lösung von (Entscheidungs-)Problemen unter Beachtung subjektiver Ziele.“ Ein Problem kann dabei als Abweichung eines aktuellen oder erwarteten Zustands von einem gewünschten Zustand angesehen werden, wobei die Aufgabe der Planung ist, geeignete Maßnahmen zu definieren, um sich dem gewünschten Zustand anzunähern (vgl. Scholl (2001, S. 7 f.)).

Das Problem, das in der vorliegenden Arbeit untersucht wird, ist folgendes: Wir nehmen an, es stehe ein (Gesamt-)Budget für die Instandhaltungsmaßnahmen eines modularen Systems über eine bestimmte Anzahl an Perioden hinweg zur Verfügung. Diese Anzahl der Perioden bezeichnen wir als Planungshorizont. Die einzelnen Komponenten des Systems unterliegen unabhängig voneinander in jeder Periode einem stochastischen Verschleiß, dessen kontinuierliche stochastische Verteilung bekannt ist oder zumindest simuliert werden kann. Weiter wird zum Ende jeder Periode eine Bewertung der Systemleistung festgestellt, die sich über eine sogenannte Strukturbewertungsfunktion (s. Abschnitt 4.3.2) aus den prozentualen Abnutzungsvorräten der Komponenten berechnen lässt. Ziel ist es, die minimale Perioden-Systemleistung im Planungszeitraum zu maximieren. Gesucht ist die Menge an (nichtnegativen) Investitionen in Instandhaltungsmaßnahmen, die jeweils zu Beginn der Perioden für die einzelnen Komponenten aufgewandt werden sollten, wobei das Gesamtbudget nicht überschritten werden darf. Die logische Abfolge von Investitionsentscheidung, Verschleiß und Bewertung der Systemleistung ist in Abb. 2.4 illustriert.

Um das beschriebene Planungsproblem zu lösen, möchten wir dieses nun einordnen und daraus sinnvolle Planungsansätze entwickeln, die im späteren Verlauf der Arbeit mathematisch formuliert, implementiert und anschließend in einer experimentellen Performance-Analyse bewertet werden.

Bezüglich der Reichweite der Planung ist das vorliegende Problem der taktischen Planung zuzuordnen. Zum einen wird das Problem typischerweise für einen mittelfristigen Zeitraum von mehreren Monaten bis wenigen Jahren betrachtet, zum anderen müssen durch die strategische Planung Rahmendaten wie die Systemstruktur oder auch die zu optimierenden Zielsetzung vorgegeben werden. Gegen die Einordnung als operative

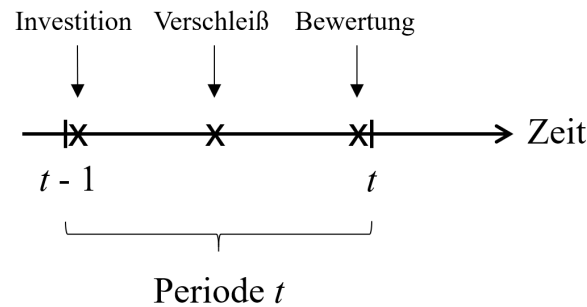


Abb. 2.4: Logische Ereignisabfolge von Investitionsentscheidung, Verschleiß und Bewertung der Systemleistung  
Quelle: Eigene Darstellung

Planung spricht, dass keine Instandhaltungsdauern berücksichtigt werden. Zudem werden in der operativen Planung aufgrund der geringeren Fristigkeit und detaillierteren Aufgabenzerlegung vorwiegend deterministische Fragestellungen behandelt (vgl. Scholl (2001, S. 15)).<sup>5</sup>

Zur Vorbereitung der Lösung des Planungsproblems werden wir ein Optimierungsmodell formulieren. Entscheidungs- und Optimierungsmodelle werden zur formalen Darstellung von Entscheidungsproblemen verwendet. Man spricht von einem Entscheidungsmodell, wenn die Menge von Alternativen explizit gegeben ist. Im Gegensatz dazu spricht man von einem (mathematischen) Optimierungsmodell, wenn die Alternativen implizit durch Nebenbedingungen gegeben sind (vgl. Scholl (2001, S. 17)). Ziel ist es jeweils, die beste Alternative bezüglich einer Zielfunktion auszuwählen.

Nach Scholl (2001, S. 19 f.) ist häufig ein bestimmter Teil der Informationen, die einem Modell zugrunde liegen, unsicher. Bei expliziter Darstellung der Unsicherheit durch Zufallsvariablen handelt es sich um ein stochastisches Modell. Vernachlässigt man diese Unsicherheit, so spricht man von einem deterministischen Modell. Bei einer skalaren Zielfunktion kann dann i. d. R. eine optimale Lösung bestimmt werden, bei der alle Festlegungen bereits zum Planungszeitpunkt erfolgen. Bei stochastischen Modellen ist dies nicht ohne Weiteres möglich. Aufgrund des stochastischen Charakters des Verschleißes in unserer Problemstellung werden wir darauf abzielen, ein stochastisches Optimierungsmodell zu formulieren.

Zudem unterscheidet man bei Modellen typischerweise noch zwischen statischen und

<sup>5</sup>Eine Interpretation als aggregiertes operatives Planungsproblem kann ebenfalls vertreten werden.

dynamischen Modellen. Wenn keine zeitliche Entwicklung vorliegt oder nicht berücksichtigt werden soll, können statische Modelle verwendet werden. Auch wenn das zugehörige reale System nach kurzer Zeit von einem Anfangszustand in einen stabilen Gleichgewichtszustand übergehen kann, kann für diesen Gleichgewichtszustand ein statisches Modell sinnvoll sein (vgl. Scholl (2001, S. 20)). Da ein stabiler Gleichgewichtszustand bei dem in Abschnitt 2.4 angesprochenen Gamma-Prozess aufgrund der zeitabhängigen Approximation des Erwartungswerts nicht erreicht wird, werden wir ein dynamisches Modell aufstellen. Die Zeitführung dieses dynamischen Modells wird diskret gewählt, um feste Planungszeitpunkte zu definieren und u. a. den Planungsansatz der stochastischen dynamischen Programmierung verfolgen zu können.

Wir möchten nun näher auf die beiden Planungsansätze eingehen, die wir für die vorliegende Problemstellung als geeignet identifiziert haben, nämlich die rollierende Planung und eine neuere Form der flexiblen Planung mit Hilfe neuronaler Netze. Diese beiden Planungsansätze werden wir in späteren Kapiteln auf das Problem anwenden und zudem in einer experimentellen Performance-Analyse miteinander vergleichen.

### 2.5.1 Rollierende Planung

Aufgrund von dynamischen und nicht vollständig vorhersehbaren Veränderungen eines zu planenden Systems und wegen der damit einhergehenden unvollständigen oder unsicheren Informationen sowie zeitlich-vertikalen Interdependenzen werden nicht sämtliche Entscheidungen schon zu Beginn einer Planung festgelegt (vgl. Scholl (2001, S. 32)).<sup>6</sup> Zeitlich-vertikale Interdependenzen liegen vor, wenn Wechselwirkungen zwischen Entscheidungen und damit verbundenen Handlungen auftreten, die zu unterschiedlichen Zeitpunkten geplant oder realisiert werden. In dem uns vorliegenden Problem liegen zeitlich-vertikale Interpendenzen vor, da bspw. die Investitionsmaßnahmen in Komponenten einer Periode zu einem geringeren Restbudget in den folgenden Perioden führen.

Rollierende Planung bedeutet, dass die Planung regelmäßig mit einem sich verschiebenden Planungszeitraum wiederholt wird, um die verbesserte Informationslage zu späteren Zeitpunkten auszunutzen. Es findet zwar eine mehrperiodige Planung statt, die Entscheidungen werden aber nur für wenige (früheste) Perioden, in unserem Fall sogar nur für

---

<sup>6</sup>Die Form der Planung, bei der zu Beginn für alle Perioden nicht mehr veränderbare Entscheidungen getroffen werden (unabhängig von der Entwicklung der Umwelt), bezeichnet man als starre Planung.

jeweils die erste Periode festgelegt. Im Anschluss wird der Planungszeitraum verschoben und zum nächsten Planungszeitpunkt mit den veränderten Bedingungen geplant. Im Gegensatz zur sog. Anschlussplanung, bei der erst nach Ende des Planungszeitraums erneut geplant wird, besteht der Hauptvorteil der rollierenden Planung darin, dass immer wieder die Gelegenheit besteht, den sich stets verbessernden Informationsstand in die Planung einfließen zu lassen (vgl. Scholl (2001, S. 139)). Mit den besseren Reaktionsmöglichkeiten einer rollierenden Planung einher geht jedoch auch eine erhöhte Planungsnervosität, da vorläufige Entscheidungen für folgende Perioden in aufeinanderfolgenden Planungszyklen häufig revidiert werden müssen.

Aufgrund der positiven Eigenschaften im Umgang mit zeitlich-vertikalen Interdependenzen und der Möglichkeit, schnell auf Abweichungen von erwarteten Umweltentwicklungen zu reagieren, soll der Planungsansatz der rollierenden Planung in der vorliegenden Arbeit weitergehend untersucht werden. Zu diesem Zweck formulieren wir zunächst ein deterministisches Ersatzwertmodell, konkret ein Erwartungswertmodell. Bei einem deterministischen Ersatzwertmodell wird jede stochastische Größe in Zielfunktion und Nebenbedingungen durch einen deterministischen Wert ersetzt; im Erwartungswertmodell verwendet man für jede stochastische Größe somit den Erwartungswert (vgl. Scholl (2001, S. 74)). Die Untersuchung dieses Planungsansatzes erfolgt in den Kapiteln 4 und 5 dieser Arbeit sowie im Rahmen einer Performance-Analyse in Kapitel 7.

### 2.5.2 Flexible Planung mit Hilfe neuronaler Netze

Als zweiter möglicher Planungsansatz zur Lösung des dargestellten Problems wird in dieser Arbeit eine besondere Form der flexiblen Planung gewählt. Allgemein kann die flexible Planung bei Vorliegen mehrstufiger Entscheidungsprozesse unter Unsicherheit eingesetzt werden. Eine flexible Planung hat zum Ziel, für jede mögliche Umweltentwicklung (Szenario), also in unserem Fall für jeden realisierbaren zeitlichen Verschleißverlauf der Komponenten, einen optimalen Plan zu ermitteln, der Eventualplan genannt wird. Ein Nachteil der flexiblen Planung ist wie bei der rollierenden Planung eine erhöhte Planungsnervosität. Bei einer begrenzten Anzahl von Szenarien, kann i. d. R. ein stochastisches Problem mit Hilfe eines sog. stochastischen Entscheidungsbaums modelliert werden, der Entscheidungsknoten für die Handlungsalternativen und Zufallsknoten für mögliche Umweltentwicklungen enthält (s. Scholl (2001, S. 58 ff.)).

Wollte man für das vorliegende Problem der Budgetallokation eine solche Entscheidungsbaumanalyse durchführen, so müsste man die möglichen (Umwelt-)Zustände und Handlungsalternativen diskretisieren, also diskrete Teilmengen aus den bei uns vorliegenden überabzählbaren Aktions- und Zustandsmengen gewinnen. Ist die Diskretisierung in Teilmengen zu grob, so kann dies zu schlechten Entscheidungen führen. Ist die Diskretisierung zu feingranular, so ist die Hauptproblematik nach Powell (2011, S. 3 ff.) allerdings die Bewältigung der drei *Flüche der Dimensionalität*:

*Zustandsraum* Wenn eine Zustandsvariable  $J$  verschiedene Dimensionen besitzt und diese wiederum jeweils  $L$  verschiedene Werte annehmen können, dann hat diese Zustandsvariable bis zu  $L^J$  verschiedene Zustände. Für den Zustandsraum in dieser Arbeit hat der Zustandsvektor mindestens eine Dimension in Höhe der Anzahl der verschiedenen Komponenten, und  $L$  wäre die Anzahl der Teilmengen, in die man den prozentualen Abnutzungsvorrat einteilt.

*Ergebnisraum der Zufallsvariablen* Die Dimensionen des Vektors der Zufallsvariablen für den Verschleiß entspricht der Anzahl der Komponenten multipliziert mit der Anzahl der Zeitschritte (wir nennen diese multiplikative Größe zur Vereinfachung  $N$ ). Wenn die Realisationen jeweils  $M$  Werte annehmen können, umfasst der Ergebnisraum  $M^N$  mögliche Ergebnisse. Bei dem angesprochenen Gamma-Verschleißprozess wäre dies für eine Diskretisierung der Wertemenge zu berücksichtigen.

*Aktionenraum* Der Entscheidungsvektor für die Investitionen hat ebenfalls  $N$  Dimensionen. Wenn eine Investition nun  $K$  verschiedene Beträge einsetzen kann, dann ergeben sich  $K^N$  mögliche Ergebnisse.

Durch diese sog. Explosion der Zustands-, Ergebnis- oder Aktionenräume können exakte Lösungen schon bei kleineren Dimensionen oder Wertebereichen von nur einem der drei Räume leicht die vorhandenen Rechnerkapazitäten übersteigen. Zudem besteht bei einer Diskretisierung auch die Möglichkeit, dass keine der als relevant erachteten Umweltentwicklungen tatsächlich eintritt, wodurch ggf. eine Neuplanung erfolgen müsste (vgl. Scholl (2001, S. 159)).

In der vorliegenden Arbeit wären alle genannten Flüche der Dimensionalität relevante, weswegen wir einen Planungsansatz der flexiblen Planung verfolgen, der direkt mit kontinuierlichen Variablen in allen drei Räumen (Zustands-, Aktionenraum und Wer-

tebereiche der Zufallsvariablen) arbeitet. Hier wird anstelle eines Entscheidungsbaums der Ansatz verfolgt, eine Funktion zu approximieren, die für einen beliebigen zulässigen Zustand der Komponenten und des Restbudgets zu jedem Zeitpunkt eine möglichst gute Investitionsentscheidung liefert, die zudem stochastische Schwankungen antizipiert.

Da jede beliebige nichtlineare Funktion über ein neuronales Netz approximiert werden kann, werden wir wie in späteren Kapiteln beschrieben ein (tiefes) neuronales Netz über den Mechanismus des bestärkenden Lernens (sog. „Deep Reinforcement Learning“) trainieren, um eine solche Funktion anzunähern. Zu diesem Zweck werden wir in Kapitel 6 das stochastische Modell formulieren und die ausgewählte Methode tiefer diskutieren und in Kapitel 7 den Planungsansatz der rollierenden Planung mit dieser Variante der flexiblen Planung vergleichen sowie auch beide Varianten einer zufälligen Planung gegenüberstellen.

# Kapitel 3

## 3 Forschungsstand

Um die vorliegende Arbeit einzuordnen, diskutieren wir im Folgenden den Stand der Forschung. Dazu wird in Abschnitt 3.1 zunächst das Vorgehen bei der Recherche und der Auswertung der Arbeiten beschrieben. Abschnitt 3.2 liefert dann einen Überblick über die wichtigsten Arbeiten und führt zur Forschungslücke, zu deren Schließung die vorliegende Dissertation einen Beitrag leistet.

### 3.1 Vorgehen bei der Recherche und Auswertung

In Tab. 3.1 werden die Ergebnisse einer quantitativen Erhebung relevanter Publikationen dargestellt. Die Analyse wurde mit der Suchmaschine „Web of Science“ durchgeführt. Im Bereich der Budgetallokation in der Instandhaltung modularer Systeme zählt die Suchmaschine 125 relevante Veröffentlichungen.

Suchbegriffe	1.	2.	3.	4.
Anzahl Treffer	5.475.200	1.585.955	38.112	1.054.068
	258.807			
		3.165		
	873			
	125			

Tab. 3.1: Übersicht über die verwendeten Suchbegriffe und Zählung der relevanten Arbeiten (Stichtag 17.02.2021)

Quelle: Eigene Darstellung



Es wurden für die Suchbegriffe Kombinationen aus vier Suchbereichen gebildet,<sup>1</sup> wobei deutsche Suchbegriffe keine relevanten Ergebnisse lieferten.

1. modular ∨ multi\*unit ∨ multi-unit ∨ multi\*component ∨ multi-component ∨ series ∨ parallel ∨ k\*of\*n ∨ k-of-n ∨ k-out-of-n ∨ components ∨ multiple ∨ network-level ∨ network\*level ∨ multi-asset ∨ multi\*asset ∨ multi-element ∨ multi\*element ∨ „system dependence“
2. maintenance ∨ maintain ∨ repair
3. budget-allocation ∨ budget\*allocation ∨ fund-allocation ∨ fund\*allocation ∨ resource-allocation ∨ resource\*allocation ∨ budget-assignment ∨ budget\*assignment ∨ „infrastructure management“ ∨ „asset management“ ∨ „financial constraint“
4. deteriorat\* ∨ degrad\*

Von den Suchergebnissen wurden themenfremde Gebiete wie etwa Medizin ausgeschlossen. Ergänzend durchgeführt wurde eine Google-Scholar-Suche mit dem Suchstring (*~maintenance OR ~instandhaltung OR ~repair*) AND (*~"budget allocation" OR ~"budgetary allocation" OR ~"budget planning" OR ~"budget allotment" OR ~"budget assignment" OR ~"budget distribution" OR ~"budget breakdown"*) AND (*~multi-unit OR ~multi-component OR ~modular OR ~series OR ~parallel OR ~k-of-n*). Die insgesamt 148 relevanten Ergebnisse seit 2016<sup>2</sup> (Stichtag 17.02.2021) wurden eingehend gesichtet. Zudem wurden folgende Überblicksartikel auf weitere Quellen hin durchgesehen: Alaswad und Xiang (2017); Cao u. a. (2018); Cho und Parlar (1991); Dekker u. a. (1997); Ding und Kamaruddin (2015); France-Mensah und O'Brien (2018); Kang u. a. (2019); Mohammadi u. a. (2019); Nicolai und Dekker (2008); Olde Keizer u. a. (2017); Petchrompo und Parlikad (2019); Ruschel u. a. (2017); Sakib und Wuest (2018); Thomas (1986); Torres-Machí u. a. (2014); Wang (2002); Wang und Chen (2016); Wu u. a. (2012).

Nach Ausschluss von Arbeiten, die keine (explizite) Budgetbegrenzung betrachten, blieben 43 für diese Dissertation einschlägige Veröffentlichungen übrig, die im folgenden Abschnitt 3.2 näher beschrieben werden.

<sup>1</sup>Der Bezeichner ∨ stellt eine ODER-Verknüpfung dar, der Stern „\*“ ist ein sog. Wildcard-Operator, wobei Bindestriche nicht durch diesen Wildcard-Operator erfasst werden (s. Clarivate Analytics (2020)).

<sup>2</sup>Es wurden nur die zurückliegenden fünf Jahre berücksichtigt, um die Suche auf aktuelle Ergebnisse einzuschränken.

## 3.2 Überblick über den Stand der Forschung

Um eine bessere Übersicht zu bekommen, grenzen wir die als relevant identifizierten Arbeiten nach Form der untersuchten Instandhaltungsmodelle voneinander ab. In Abschnitt 3.2.1 werden Modelle zur mehrperiodigen Budgetallokation bei monolithischen Systemen<sup>3</sup> behandelt. Abschnitt 3.2.2 untersucht Modelle, die sich mit der Budgetaufteilung auf mehrere Komponenten befassen, aber nicht die Verteilung über mehrere Perioden untersuchen. Abschnitt 3.2.3 behandelt komplexere Instandhaltungsmodelle, die sowohl Modularität als auch Mehrperiodigkeit berücksichtigen und damit dem Modell, das in dieser Dissertation ausführlich behandelt wird, am nächsten kommen. Abb. 3.1 ordnet den Forschungsbereich dieser Dissertation in die Landschaft der unterschiedlichen Instandhaltungsmodelle ein. Periodenbezogene Budgetallokation bedeutet in diesem Zusammenhang, dass trotz eines mehrperiodigen Zeitraums eine strikte Budgetbegrenzung für jede einzelne Periode vorgegeben ist. Andere Arbeiten betrachten ein Gesamtbudget für ihr mehrperiodiges Instandhaltungsmodell, also eine periodenübergreifende Budgetallokation. Für eine bessere Einordnung und Übersichtlichkeit der Quellen ist in Tab. 3.2 eine Klassifikation für die Ausprägungen der einzelnen Kriterien dargestellt, die wir in den folgenden Unterabschnitten immer wieder verwenden.

### 3.2.1 Budgetallokation bei monolithischen, mehrperiodigen Instandhaltungsmodellen

Drei relevante Arbeiten betrachten Budgetallokation bei monolithischen, mehrperiodigen Instandhaltungsmodellen. Dabei untersuchen die Arbeiten von Swei u. a. (2019) und Wang u. a. (2007) Problemstellungen, bei denen das Budget periodenbezogen gegeben ist. Die Arbeit von Liu und Frangopol (2005) hingegen betrachtet eine periodenübergreifende Budgetallokation. Tab. 3.3 liefert eine alphabetisch nach Autoren sortierte Übersicht der Arbeiten und klassifiziert sie jeweils gemäß den in Tab. 3.2 aufgeführten Ausprägungen.

---

<sup>3</sup>Monolithische Systeme sind einkomponentige Systeme oder modulare Systeme, deren Komponenten in unserem Kontext wie eine einzige behandelt werden.

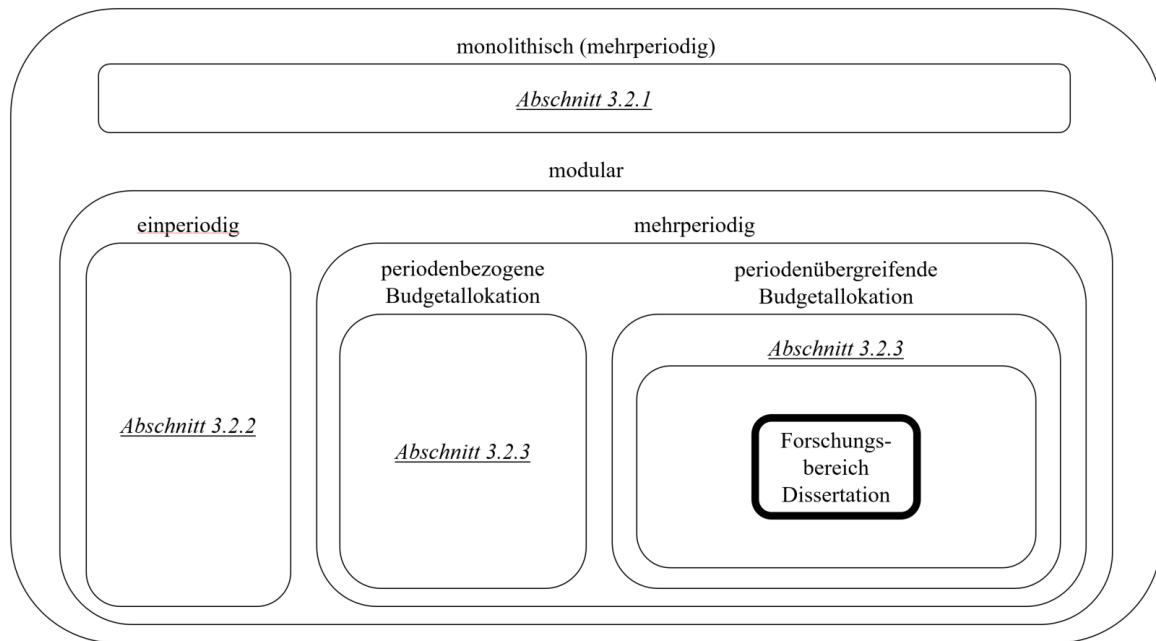


Abb. 3.1: Budgetallokation in der Instandhaltung – Forschungsbereich der Dissertation  
Quelle: Eigene Darstellung

### 3.2.2 Budgetallokation bei modularen, einperiodigen Instandhaltungsmodellen

Eine Reihe von Autoren untersuchen modulare, einperiodige Instandhaltungsmodelle. Bei der Budgetallokation von einperiodigen Instandhaltungsmodellen spielt Verschleiß eine untergeordnete Rolle. Es wurden 17 Arbeiten identifiziert, die sich mit dieser Form der Instandhaltungsmodellierung befassen. Unter diesen Arbeiten nimmt nur jene von Chowdhury und Tan (2005) binäre Zustände bei den Komponenten an, die übrigen befassen sich mit Multizustandssystemen. Von diesen wiederum behandeln folgende Arbeiten abzählbar viele Zustände: Dehghani und Sherali (2016); Fwa u. a. (2000); Fwa und Farhan (2012); Gao und Zhang (2013); Nahas und Nourelfath (2005); Šelih u. a. (2008). Überabzählbar viele Zustände – wie es auch in der vorliegenden Dissertation der Fall ist – sind Teil der folgenden Arbeiten: Chan u. a. (2003); Dehghani u. a. (2013); Gabriel u. a. (2006); Menendez u. a. (2013); Patidar u. a. (2011); Roshandeh u. a. (2015); Sadek u. a. (2003); Truong u. a. (2018); Yadollahi u. a. (2015). Instandhaltungsdauern

Dimension	Ausprägungen
Budgetrestriktion	1) perioden-bezogen, 2) periodenübergreifend
Planungshorizont	1) einperiodig, 2) endlich, 3) unendlich
Art der Zeitführung	1) statisch, 2) dynamisch diskret, 3) dynamisch stetig
Systemstruktur	1) monolithisch, 2) modular
Anzahl der Zustände	1) binär, 2) abzählbar, 3) überabzählbar
Bestimmtheitsgrad der Zustandsänderungen (Verschleiß)	1) deterministisch, 2) stochastisch
Instandhaltungsdauern	1) vernachlässigbar, 2) berücksichtigt
Ziele der Instandhaltung	1) Minimierung der Gesamtkosten bzw. Gesamtkostenrate, 2) Maximierung der Systemverfügbarkeit, 3) Maximierung der Systemleistung, 4) sonstige einkriterielle Zielsetzung, 5) multikriterielle Zielsetzung

Tab. 3.2: Klassifikation zu Instandhaltungsmodellen

Quelle: Eigene Darstellung

Arbeit	Budget- restrik- tion	Planungs- horizont	Art der Zeitfüh- rung	System- struktur	Anzahl der Zu- stände	Ver- schleiß	Instand- hal- tungs- dauern	Ziele der Instand- haltung
Liu und Frangopol (2005)	2	2	3	1	3	1, 2	1	5
Swei u. a. (2019)	1	2	2		3	2	1	3
Wang u. a. (2007)	1	2	2		2	2	1	5

Tab. 3.3: Einordnung der Arbeiten zu Budgetallokation bei monolithischen, mehrperiodi-  
gen Instandhaltungsmodellen

Quelle: Eigene Darstellung

werden in diesen Arbeiten normalerweise vernachlässigt, nur Fwa u. a. (2000) zieht sie in die Modellierung explizit mit ein. Eine alphabetisch sortierte Übersicht über die Arbeiten des Abschnitts liefert Tab. 3.4 auf der nächsten Seite.

### **3.2.3 Budgetallokation bei modularen, mehrperiodigen Instandhaltungsmodellen**

Insgesamt wurden 23 relevante Arbeiten für die Budgetallokation in modularen, mehrperiodigen Instandhaltungsmodellen identifiziert. Davon behandeln 19 Arbeiten ausschließlich eine periodenbezogene Budgetallokation, also eine Budgetallokation mit Budgetrestriktionen für jede einzelne Periode. Nur vier Arbeiten behandeln eine auch in der vorliegenden Dissertation angenommene periodenübergreifende Budgetallokation.

#### **Periodenbezogene Budgetverteilung**

Von den 19 Arbeiten, die eine periodenbezogene Budgetverteilung zugrunde legen, gehen neun von einem deterministischen Verschleiß aus: Ameri u. a. (2019); de la Garza u. a. (2011); Gao u. a. (2012); Grussing und Liu (2014); Hackl u. a. (2018); Meegoda u. a. (2009); Ouyang und Madanat (2004); Pantelias und Zhang (2009); Wang u. a. (2003). Einen stochastischen Verschleiß nehmen zehn Arbeiten an: Chootinan u. a. (2006); Kuhn (2010, 2012); Medury und Madanat (2013, 2014); Mild und Salo (2009); Mohammadi u. a. (2020); Morcoux und Lounis (2005); Sarkale u. a. (2018); Shi u. a. (2021).

Einen alphabetisch sortierten Überblick über die Arbeiten, bei denen eine periodenbezogene Budgetverteilung bei modularen, mehrperiodigen Instandhaltungsmodellen vorgenommen wird, gibt Tab. 3.5 auf Seite 30.

#### **Periodenübergreifende Budgetverteilung**

Die vier in der Recherche identifizierten Arbeiten, die in den in Abb. 3.1 auf Seite 26 angezeigten Forschungsbereich der vorliegenden Arbeit fallen, werden wir nun eingehender betrachten, und zwar jeweils hinsichtlich 1) Problemstellung und Anwendungsgebiet, 2) spezifischer Eigenschaften des Planungsproblems, 3) Planungsansatz und Lösungsmethodik, 4) Ergebnissen sowie 5) betrachteter Instanzen. Die Nennungen folgen dabei in alphabetischer Reihe.

Arbeit	Budget- restrik- tion	Planungs- horizont	Art der Zeitfüh- rung	System- struktur	Anzahl der Zu- stände	Ver- schleiß	Instand- hal- tungs- dauern	Ziele der Instand- haltung
Chan u. a. (2003)	2	1	1	2	3	—	1	5
Chowdhury und Tan (2005)					1	—	1	5
Dehghani u. a. (2013)					3	—	1	4
Dehghani und Sherali (2016)					2	2	1	4
Fwa u. a. (2000)					2	—	2	5
Fwa und Farhan (2012)					2	—	1	1, 3
Gabriel u. a. (2006)					3	—	1	5
Gao und Zhang (2013)					2	—	1	3
Gharaibeh u. a. (2006)					3	—	1	3
Menendez u. a. (2013)					3	—	1	3
Nahas und Nourelfath (2005)					2	—	1	2
Patidar u. a. (2011)					3	—	1	1
Roshandeh u. a. (2015)					3	—	1	5
Sadek u. a. (2003)					3	—	1	3
Šelih u. a. (2008)					2	—	1	5
Truong u. a. (2018)					3	—	1	5
Yadollahi u. a. (2015)					3	—	1	5

Tab. 3.4: Einordnung der Arbeiten zu Budgetallokation bei modularen, einperiodigen Instandhaltungsmodellen

Quelle: Eigene Darstellung

Arbeit	Budget- restrik- tion	Planungs- horizont	Art der Zeitfüh- rung	System- struktur	Anzahl der Zu- stände	Ver- schleiß	Instand- hal- tungs- dauern	Ziele der Instand- haltung
Ameri u. a. (2019)	1	2	2	2	2	1	1	1
Chootinan u. a. (2006)		2	2		3	2	1	3
de la Garza u. a. (2011)		2	2		2	1	1	4
Gao u. a. (2012)		2	2		2	1	1	5
Grussing und Liu (2014)		2	3		2	1	1	5
Hackl u. a. (2018)		2	2		2	1	2	1
Kuhn (2010)		2	2		2	2	1	1
Kuhn (2012)		2	2		2	2	1	1 – 4
Medury und Madanat (2013)		2	2		2	2	1	1
Medury und Madanat (2014)		2	2		2	2	1	1
Meegoda u. a. (2009)		2	2		2	1	1	1
Mild und Salo (2009)		2	2		2	2	1	4
Mohammadi u. a. (2020)		2	2		3	2	1	3
Morcous und Lounis (2005)		2	2		2	2	1	1, 3
Ouyang und Madanat (2004)		2	3		3	1	1	1
Pantelias und Zhang (2009)		2	2		3	1	1	4
Sarkale u. a. (2018)		2	2		2	2	2	4
Shi u. a. (2021)		2	2		2	2	1	3
Wang u. a. (2003)		2	2		3	1	1	1, 3

Tab. 3.5: Einordnung der Arbeiten zu Budgetallokation bei modularen, mehrperiodigen Instandhaltungsmodellen mit periodenbezogener Budgetallokation  
Quelle: Eigene Darstellung

Lee und Madanat (2015) behandeln die gemeinsame Optimierung von Sanierungs- und Erneuerungsmaßnahmen für den Straßenbelag von großen Straßensystemen. Ihr Ziel ist es, die Gesamtkostenrate für ein betrachtetes System zu minimieren. Ihre Menge an Handlungsoptionen ist dabei diskret. Als Planungsansatz bedienen sie sich starrer hierarchischer Planung. Zur Lösung nutzen sie auf Systemebene einen genetischen Algorithmus, auf Ebene der einzelnen Straßensegmente<sup>4</sup> werden mathematische Programme gelöst. Im Ergebnis zeigen sie die wirtschaftliche Überlegenheit eines kombinierten Budgets für Sanierungs- und Erneuerungsmaßnahmen im Vergleich zu getrennten Budgets für einzelne Bereiche. Als Instanz betrachten sie einen Teil des kalifornischen Highway-Systems.

Nozhati u. a. (2019) identifizieren Instandhaltungsmaßnahmen zur Wiederherstellung eines Stromnetzes nach einem schweren Erdbeben. Ihr Planungsproblem vernachlässigt dabei Verschleißprozesse, stattdessen werden Instandhaltungsdauern berücksichtigt. Ihr Planungsansatz ist der starren hierarchischen Planung zuzurechnen. Sie verfolgen einen iterativen Optimierungsansatz hinsichtlich zweier korrelierter Ziele, die sie unabhängig voneinander, aber mit den gleichen Methoden optimieren: 1) Minimierung der Anzahl an Tagen, um einen bestimmten Anteil der Bevölkerung einer Gemeinde wieder mit Strom zu versorgen und 2) Maximierung der Anzahl an mit Strom versorgten Personen pro Zeiteinheit. Sie nutzen approximative dynamische Programmierung und andere Heuristiken, um der „Explosion“ ihres abzählbaren Zustandsraum (vgl. Abschnitt 2.5.2) entgegenzuwirken. Damit erreichen sie annähernd Optimalität bezüglich der von ihnen gewählten Zielfunktionen. Ihre Berechnungen haben sie auf einer Testumgebung durchgeführt, die in Anlehnung an die Gemeinde Gilroy, Kalifornien, USA generiert wurde.

Sathaye und Madanat (2011) stellen einen Ansatz zur Optimierung der Belagserneuerung von Autobahnen vor. Sie möchten die optimale Häufigkeit und Intensität der Durchführung von Instandhaltungsmaßnahmen bestimmen. Neben der Betrachtung von rein deterministischem Verschleiß analysieren die Autoren die Sensitivität bzgl. der Unsicherheit der Parameter für ihre Verschleißmodellierung. Sie führen eine starre Planung durch und nehmen einen unendlichen Planungshorizont an, da sie davon ausgehen, dass eine stationäre Lösung innerhalb einer kurzen Dauer erreicht wird. Die Autoren lösen ein nichtlineares Optimierungsproblem mit Hilfe eines eigens entwickelten Algorithmus, der Konvexitäts-Eigenschaften ihrer Modellierung ausnutzt. Dadurch finden sie die optimale Lösung in einem aus drei Komponenten bestehenden Beispiel-System.

---

<sup>4</sup>Ein Segment entspricht einer Komponente des Systems.



In einer späteren Arbeit (Sathaye und Madanat (2012)) stellen die Autoren einen weiteren Ansatz zur Optimierung der Belagserneuerung von Straßen vor, diesmal explizit für große Systeme. Auch in dieser Arbeit verfolgen sie das Ziel, die optimale Häufigkeit und Intensität der Durchführung von Instandhaltungsmaßnahmen zu bestimmen. Eine Besonderheit der Arbeit ist, dass viele Modellparameter inkl. derjenigen für den Verschleiß aus Wahrscheinlichkeitsverteilungen gezogen werden. Prinzipiell wird für die einzelnen Instanzen weiterhin ein deterministischer Verschleiß angenommen. Sie begründen ihre starre Planung erneut mit der Annahme, dass eine stationäre Lösung innerhalb einer kurzen Dauer erreicht wird. Um ihre Instanzen zu lösen, formulieren die Autoren das duale Problem ihres nichtlinearen Optimierungsproblems und wenden ein Gradientenabstiegsverfahren in Kombination mit dem Newton-Raphson-Verfahren an. Dadurch finden sie die optimale Lösung in einem Beispiel-System, welches aus 1000 Komponenten besteht.

Ein Überblick dieser Arbeiten hinsichtlich der in Tab. 3.2 dargestellten Klassifikation findet sich in Tab. 3.6.

Arbeit	Budget- restrik- tion	Planungs- horizont	Art der Zeitfüh- rung	System- struktur	Anzahl der Zu- stände	Ver- schleiß	Instand- hal- tungs- dauern	Ziele der Instand- haltung
Lee und Madanat (2015)	2	3	3	2	3	1	1	1
Nozhati u. a. (2019)		2	2		2	—	2	5
Sathaye und Madanat (2011)		3	3		3	1	1	1
Sathaye und Madanat (2012)		3	3		3	1	1	1

Tab. 3.6: Einordnung der Arbeiten zu Budgetallokation bei modularen, mehrperiodigen Instandhaltungsmodellen mit periodenübergreifender Budgetallokation  
Quelle: Eigene Darstellung

### 3.2.4 Forschungslücke

Es werden in der Literatur zahlreiche Ansätze für die Budgetallokation in der Instandhaltung diskutiert. Bei den in Abschnitt 3.2.1 aufgeführten Arbeiten wird vernachlässigt, dass ein System aus mehreren Komponenten bestehen kann, die sich unterschiedlich verhalten und somit auch die komponentenspezifischen Instandhaltungsausgaben bestimmt werden sollten. Die in Abschnitt 3.2.2 genannten Arbeiten berücksichtigen nicht ausreichend, dass Instandhaltungsmaßnahmen, die auf dasselbe Budget zugreifen, zu unterschiedlichen Zeitpunkten anfallen können. Demgegenüber beschäftigt sich die vorliegende Arbeit mit Budgetallokation in einem modularen, mehrperiodigen Instandhaltungsmodell wie auch die in Abschnitt 3.2.3 genannten Autoren.

Viele dieser Autoren haben bereits die Frage untersucht, wie eine Budgetallokation in einem modularen, mehrperiodigen Modell erfolgen sollte, wenn es eine strikte periodenbezogene Vorgabe für das Budget gibt (vgl. Tab. 3.5). Allerdings ist es vor allem aus ökonomischer Sicht sinnvoll, ein Gesamtbudget für den aus mehreren Perioden bestehenden gesamten Planungszeitraum festzulegen, da dann der Optimierungsspielraum größer ist. Wenn es keine festen Vorgaben für die einzelnen zu planenden Perioden gibt, ist ein besseres Gesamtergebnis möglich. Den Umfang der möglichen Verbesserung werden wir im Rahmen der experimentellen Analyse in Abschnitt 7.2.1 untersuchen.

In dieser Arbeit wird der Ansatz einer periodenübergreifenden Budgetallokation in einem modularen System verfolgt, wie es auch bei den vier dazu identifizierten Arbeiten des Abschnitts 3.2.3 der Fall ist. Der Beitrag der Dissertation im Vergleich zu den genannten Untersuchungen aus Abschnitt 3.2.3 bezieht sich neben der gewählten Methodik zur Lösung des Planungsproblems insbesondere auf die Verallgemeinerung der Annahmen, die diese Autoren für die Verschleißmodellierung zugrunde legen.

So setzen die Arbeiten von Lee und Madanat (2015) sowie von Nozhati u. a. (2019) keinen Schwerpunkt auf die präzise Abbildung von Verschleißprozessen und modellieren Verschleiß daher deterministisch oder gar nicht. Die Arbeiten von Sathaye und Madanat (2011, 2012) begründen ihre Art der Modellierung: Sie gehen mit dem Hinweis auf die Arbeiten von Li und Madanat (2002) und von Ouyang und Madanat (2006) davon aus, dass sich eine stationäre Lösung bei unendlichem Planungshorizont einstellen würde. Diese Arbeiten zeigten, dass optimale Lösungen relativ unempfindlich gegenüber den Anfangsbedingungen seien und dass die stationäre Lösung früh nach der Implementierung

optimaler Politiken erreicht werde. Daher vereinfache die Lösung des Problems mit unendlichem Horizont die Methodik, ohne ihre Anwendung zu behindern. Zudem könne gefolgert werden, dass der konkrete Zeitpunkt einer Instandhaltungsmaßnahme zu vernachlässigen sei. Die Entscheidungsvariablen ihres Optimierungsmodells beziehen sich zudem auf den zeitunabhängigen stationären Zustand. Da die Lösung ihres Optimierungsmodells nur einmalig bestimmt wird, liegt eine starre Planung vor.

In der vorliegenden Arbeit hingegen gehen wir davon aus, dass die Annahme einer stationären Verteilung für die Verschleißmodellierung nicht zwangsläufig zutreffen muss. Dass es nichtstationäre Verteilungen im Bereich der Instandhaltung gibt, wurde bereits in Abschnitt 2.4 angesprochen. Daraus abgeleitet verwenden wir den Planungsansatz der rollierenden Planung und den der flexiblen Planung (mit Hilfe neuronaler Netze), wobei der Verschleiß als stochastische Zufallsvariablen modelliert wird. Weiterhin unterscheidet sich die vorliegende Arbeit von den Ansätzen der Literatur dadurch, dass wir eine allgemeine Formulierung einer Investitionsfunktion entwickeln, die auf dem abnehmenden Grenznutzen beruht und die diskretisierte Investition in Instandhaltungsmaßnahmen für eine taktische bzw. aggregierte Planung hinreichend genau annähern kann.

Zusammenfassend kann festgestellt werden, dass in der Literatur keine andere Arbeit identifiziert wurde, die Budgetallokation für ein

- *modulares* System behandelt,
- dessen Komponenten einem Verschleiß unterliegen, für den eine *nichtstationäre Verteilungsannahme* zugelassen ist,
- wenn ein Gesamtbudget *periodenübergreifend* für einen *mehrperiodigen* Planungszeitraum zur Verfügung steht.

# 4 Erwartungswertmodell für die rollierende Planung der Budgetallokation

Wir wollen nun das Erwartungswertmodell für die rollierende Planung wie in Abschnitt 2.5.1 besprochen entwickeln. Dazu werden zunächst in Abschnitt 4.1 die verwendeten Mengen, Variablen und Parameter eingeführt. Im Anschluss gehen wir in Abschnitt 4.2 auf die Verschleißmodellierung ein. Abschnitt 4.3 beschreibt dann die wichtigsten funktionalen Zusammenhänge und wie die dem Modell zugrunde liegenden Prämissen abgebildet werden. Zuletzt führt Abschnitt 4.4 diese Komponenten in dem deterministischen Erwartungswertmodell zusammen.

## 4.1 Mengen, Variablen und Parameter

Den Index für die Komponenten eines Systems bezeichnen wir im Folgenden mit  $i$ , die Anzahl der Komponenten des Systems mit  $n$ , es gilt also  $i \in \{1, \dots, n\}$ . Den Periodenindex bezeichnen wir mit  $t$ , der Planungszeitraum umfasst  $T$  Perioden, sodass üblicherweise  $t \in \{1, \dots, T\}$  gilt.<sup>1</sup> Es steht weiterhin ein positives Gesamtbudget  $\hat{B}$  zur Verfügung, das über alle Perioden und Komponenten verteilt werden kann. Das Gesamtbudget geht als externer Parameter in das Modell ein. Zur besseren Unterscheidung der deterministischen externen Parameter kennzeichnen wir diese wie in  $\hat{B}$  mit einem diakritischen Zirkumflexzeichen. Als Hilfsvariablen führen wir die Restbudgets am Ende der Perioden  $t$  ein, die wir mit  $B_t$  bezeichnen.

---

<sup>1</sup>Für die Modellierung wird zur Abbildung von Anfangswertbedingungen auch  $t = 0$  zugelassen.

Die Höhen der Instandhaltungsinvestitionen in die Komponenten  $i$  zu Beginn der Perioden  $t$  entsprechen Entscheidungsvariablen  $x_{it}$ , wobei offensichtlich gilt, dass

$$\sum_{i=1}^n \sum_{t=1}^T x_{it} \leq \hat{B} \quad (4.1)$$

erfüllt sein muss, also in Summe nicht mehr als das Gesamtbudget ausgegeben werden darf.<sup>2</sup> Ferner muss für alle  $i = 1, \dots, n$  und alle  $t = 1, \dots, T$  natürlich  $x_{it} \geq 0$  gelten.

Jede Komponente  $i$  befindet sich zum Ende der Periode  $t$  in einem Zustand  $p_{it} \geq 0$ . Präziser gesprochen bezeichnet  $p_{it}$  die kardinale Bewertung dieses Zustands, die wir bereits in Abschnitt 2.4 als den prozentualen Abnutzungsvorrat interpretiert haben. Der Wert 0 entspricht einer unbrauchbaren und der Wert 1 einer neuwertigen Komponente.

Die Anfangszustände  $p_{i0} = \hat{p}_i \in [0; 1]$  werden wie das Gesamtbudget als Anfangswertbedingungen in Form von externen Parametern vorgegeben. Der Zustand von Komponente  $i$  kann sich vom Ende der Periode  $t - 1$ , das gleichzeitig dem Beginn der Periode  $t$  entspricht, bis zum Ende der Periode  $t$  zum einen durch den Einsatz von Investitionen verbessern, zum andern durch Verschleiß in Periode  $t$  verschlechtern. Dieser Verschleiß wird im späteren Verlauf (s. Abschnitt 4.2 bzw. Kapitel 6) als Zufallsvariable modelliert. Da die Stochastizität allerdings in diesem Kapitel über den Planungsansatz der rollierenden Planung erfasst wird und wir hier nur das deterministische Erwartungswertmodell betrachten, ist vorläufig nur der erwartete Verschleiß von Interesse. Dieser erwartete Verschleiß von Komponente  $i$  in der Periode  $t$  geht als externer Parameter in das Modell ein. Wir bezeichnen ihn mit  $\widehat{\Delta w}_{it} > 0$  und wollen nun beschreiben, welche Werte dafür sinnvoll erscheinen.

## 4.2 Verschleißmodellierung

Während Verschleiß bei Multizustandssystemen mit endlich vielen Zuständen meist über Markov-Prozesse modelliert wird, nutzt man für die Modellierung überabzählbar vieler Zustände den Wiener Prozess<sup>3</sup>, den Gamma-Prozess oder den inversen Gauss-Prozess

---

<sup>2</sup>Zum besseren Verständnis der Indizierung der Variablen erinnern wir an die logische Ereignisabfolge von Investitionsentscheidung, Verschleiß und Bewertung der Systemleistung, dargestellt in Abb. 2.4 auf Seite 18.

<sup>3</sup>Der Wiener-Prozess wird in manchen Anwendungskontexten auch als *Brown'sche Bewegung* bezeichnet.

(vgl. Alaswad und Xiang (2017) bzw. Gorjian u. a. (2010)).<sup>4</sup> Wir nehmen in dieser Arbeit einen graduellen Verschleiß an, der durch kontinuierliche Nutzung entsteht. Für diese Art von Verschleiß wird in der Literatur der Gamma-Prozess als geeignet angesehen (vgl. van Noortwijk (2009)).

Die Zufallsvariable<sup>5</sup>  $\widetilde{W}$  ist *gammaverteilt* mit Formparameter  $v > 0$  und Skalierungsparameter  $u > 0$ , wenn ihre Dichtefunktion gegeben ist durch

$$\text{Ga}(w | v, u) := \frac{u^v}{\Gamma(v)} w^{v-1} \exp(-u \cdot w),$$

mit  $w \in (0, \infty)$  und Gammafunktion  $\Gamma(v) = \int_{z=0}^{\infty} z^{v-1} \exp(-z) dz$  für  $v > 0$ . Wenn nun die Formfunktion  $v(t)$  eine nichtfallende, rechtsseitig stetige, reellwertige Funktion für  $t \geq 0$  mit  $v(0) = 0$  ist, dann wird ein zeitstetiger stochastischer Prozess  $\{\widetilde{W}(t), t \geq 0\}$  mit den folgenden Eigenschaften als *Gamma-Prozess* bezeichnet.

1.  $\mathbb{P}(\widetilde{W}(0) = 0) = 1$ , d. h., zum Zeitpunkt 0 ist noch kein Verschleiß aufgetreten.
2.  $\widetilde{W}(\tau) - \widetilde{W}(t)$  ist verteilt gemäß  $\text{Ga}(v(\tau) - v(t), u) \forall \tau > t \geq 0$ , d. h., die Differenz der Zufallsvariablen zu zwei verschiedenen Zeitpunkten ist ebenfalls gammaverteilt.
3.  $\widetilde{W}(t)$  hat stochastisch unabhängige Zuwächse.

Die Zufallsvariable  $\widetilde{W}(t)$  entspricht dem *kumulierten* Verschleiß im Zeitintervall  $[0; t]$ .

Damit ist die Dichtefunktion von  $\widetilde{W}(t)$  zum Zeitpunkt  $t$  beschrieben durch

$$\text{Ga}(w | v(t), u)$$

mit Erwartungswert

$$\mathbb{E}(\widetilde{W}(t)) = \frac{v(t)}{u}$$

und Varianz

$$\sigma^2(\widetilde{W}(t)) = \frac{v(t)}{u^2}.$$

Um nun die Parameter eines Gamma-Verschleißprozesses zu schätzen, zieht man empirische Überlegungen heran. Es ist empirisch belegt, dass der Erwartungswert häufig

---

<sup>4</sup>Die Grundlagen zu stochastischen Prozessen kann man bspw. bei Webel und Wied (2016) nachlesen.

<sup>5</sup>Das diakritische Tilde-Zeichen soll hier und im weiteren Verlauf Zufallsvariablen kenntlich machen.

proportional zu einem Monom im Alter  $t$  der betrachteten Komponente ist (vgl. van Noortwijk (2009)), also

$$\mathbb{E}(\widetilde{W}(t)) = \frac{v(t)}{u} = \frac{ct^b}{u} = at^b$$

mit Konstanten  $a > 0$  (bzw.  $c > 0$ ) und  $b > 0$  (vgl. auch Abschnitt 2.4).

Sinnvolle Werte für die Parameter  $\widehat{\Delta w}_{it}$  des Erwartungswertmodells erhält man also, wenn man die Differenz von Erwartungswerten bildet:

$$\widehat{\Delta w}_{it} := \mathbb{E}(\widetilde{W}(t+1) - \widetilde{W}(t)) = \mathbb{E}(\widetilde{W}(t+1)) - \mathbb{E}(\widetilde{W}(t))$$

mit  $\widehat{\Delta w}_{it} > 0$ . Das Delta-Zeichen „ $\Delta$ “ weist darauf hin, dass der zugrundeliegende Gamma-Prozess  $\{\widetilde{W}(t), t \geq 0\}$  eine *kumulierte* Verschleißverteilung nachbildet und in das Erwartungswertmodell eine *Periodendifferenz* einfließt.

## 4.3 Modellierung der Dynamik, der Systemleistung und der Planungszielsetzung

Wir möchten nun die funktionalen Zusammenhänge des Erwartungswertmodells besprechen. Die Entwicklung des prozentualen Abnutzungsvorrats modellieren wir über eine Zustandsübergangsfunktion (Abschnitt 4.3.1), die Bewertung der Systemleistung am Ende einer Periode erfolgt über eine Strukturbewertungsfunktion (Abschnitt 4.3.2), und die Gesamtbewertung des Einsatzes der Instandhaltungsinvestitionen wird über die Zielfunktion (Abschnitt 4.3.3) des Optimierungsproblems ermittelt.

### 4.3.1 Entwicklung der Komponentenzustände

Über die Zustandsübergangsfunktion beschreiben wir, wie sich der prozentuale Abnutzungsvorrat einer Komponente vom Ende einer Periode zum Ende der Folgeperiode entwickelt. Diese Entwicklung vollzieht sich in Abhängigkeit 1) des prozentualen Abnutzungsvorrats  $p$  am Ende der Vorperiode, 2) der Investition  $x$  in Instandhaltungsmaßnahmen für die Komponente zu Beginn der Periode und 3) des erwarteten Verschleißes  $\widehat{\Delta w}$  innerhalb der Periode.

Wir nennen diese komponentenspezifische Funktion  $g_i$ , die wir folgendermaßen definieren wollen:

$$g_i : \mathbb{R}_{\geq 0} \times [0; 1] \times \mathbb{R}_{> 0} \longrightarrow [0; 1], (x, p, \Delta w) \mapsto g_i(x, p, \Delta w).$$

Es gilt also für den prozentualen Abnutzungsvorrat der Komponente  $i$  am Ende der Periode  $t$ , dass

$$p_{it} = g_i(x_{it}, p_{i,t-1}, \widehat{\Delta w}_{it}).$$

Jede geeignete Zustandsübergangsfunktion  $g_i$  muss somit sicherstellen, dass sich der prozentuale Abnutzungsvorrat der Komponente  $i$  zwischen 0 und 1 bewegt. Wir nehmen weiter an, dass Investitionen in eine Komponente und der Verschleiß dieser Komponente innerhalb einer Periode nicht voneinander abhängen. Somit können wir die Zustandsübergangsfunktion  $g_i$  in eine Investitionsfunktion

$$g_i^+ : \mathbb{R}_{\geq 0} \times [0; 1] \longrightarrow [0; 1], (x, p) \mapsto g_i^+(x, p)$$

und den negativen erwarteten Verschleiß  $-\widehat{\Delta w}_{it}$  zerlegen, d. h.,

$$g_i(x_{it}, p_{i,t-1}, \widehat{\Delta w}_{it}) = \max \left\{ g_i^+(x_{it}, p_{i,t-1}) - \widehat{\Delta w}_{it}; 0 \right\}.$$

Das Maximum mit 0 wird gebildet, da kein negativer prozentualer Abnutzungsvorrat erreicht werden kann. Für die Festlegung der Eigenschaften von  $g_i^+$  möchten wir das Gesetz des abnehmenden Grenznutzens berücksichtigen, das wir in Abschnitt 2.2.2 angesprochen haben. Es sei daran erinnert, dass wir den Nutzen einer Investition in eine Komponente über kardinale Werte abbilden, indem wir ihn mit dem prozentualen Abnutzungsvorrat assoziieren.

Sei  $x_{it}$  die Investition in die Instandhaltung der Komponente  $i$  zu Beginn der Periode  $t$  und wir nehmen an, dass die Funktionen  $g_i^+$  mindestens zweimal ableitbar nach  $x_{it}$  sind. Dann gilt für den Nutzen  $U(x_{it}) = g_i^+(x_{it}, p_{i,t-1})$ , dass für  $p_{i,t-1} < 1$

$$\frac{\partial}{\partial x_{it}} U(x_{it}) = \frac{\partial}{\partial x_{it}} g_i^+(x_{it}, p_{i,t-1}) > 0 \quad (4.2)$$



und

$$\frac{\partial^2}{\partial x_{it}^2} U(x_{it}) = \frac{\partial^2}{\partial x_{it}^2} g_i^+(x_{it}, p_{i,t-1}) < 0. \quad (4.3)$$

Dabei besagt Gl. (4.2), dass eine Investition immer einen positiven Nutzenzuwachs bewirkt, also einen positiven Grenznutzen besitzt, solange die Komponente zu Beginn der Periode  $t$  nicht über einen prozentualen Abnutzungsvorrat von 100 % verfügt. Dieser Grenznutzen wird auch als *Effektivität*  $\eta_i(x_i)$  der Investition  $x_i$  bezeichnet. Gl. (4.3) beschreibt den abnehmenden Grenznutzen der Investition, also sind die Funktionen  $g_i^+$  konkav in  $x_{it}$ . Zudem wollen wir auch die Ableitbarkeit von  $g_i^+$  nach  $p_{i,t-1}$  vereinbaren.

Da der Verschleiß  $\widehat{\Delta w}_{it}$  unabhängig von  $x_{it}$  ist, gilt somit für die Zustandsübergangsfunktion unter den Voraussetzungen  $\widehat{\Delta w}_{it} < g_i^+(x_{it}, p_{i,t-1})$  und  $p_{i,t-1} < 1$

$$\frac{\partial}{\partial x_{it}} g_i(x_{it}, p_{i,t-1}, \widehat{\Delta w}_{it}) > 0$$

und

$$\frac{\partial^2}{\partial x_{it}^2} g_i(x_{it}, p_{i,t-1}, \widehat{\Delta w}_{it}) < 0.$$

Für  $x_{it}$  mit  $g_i^+(x_{it}, p_{i,t-1}) \leq \widehat{\Delta w}_{it}$  ist  $\frac{\partial^+}{\partial x_{it}} g_i(x_{it}, p_{i,t-1}, \widehat{\Delta w}_{it}) = 0$ .

Ferner nehmen wir an, dass

$$\lim_{x_{it} \rightarrow \infty} g_i^+(x_{it}, p_{i,t-1}) = 1.$$

### 4.3.2 Strukturbewertung

Um die Einzelbewertungen der Komponentenzustände, also der prozentualen Abnutzungsvorräte  $p_{it}$  der Komponenten  $i$ , zum Ende der Periode  $t$  in einer Kennzahl zusammenzuführen, die das jeweils vorliegende modulare Gesamtsystem bewertet, definieren wir eine Strukturbewertungsfunktion

$$\varphi : [0; 1]^n \longrightarrow [0; 1], \mathbf{p}_t \mapsto \varphi(\mathbf{p}_t). \quad (4.4)$$

Die Kennzahl, die sich über die Berechnungsvorschrift  $\varphi$  ergibt, nennen wir Systemleistung, welche (wie die Komponentenzustände) zwischen 0 % und 100 % liegt.

Um sinnvolle Beispiele einer solchen Kennzahlberechnung zu finden, werfen wir einen Blick in die Literatur. Dort ist das Konzept von *Struktur*funktionen häufig im Zusammenhang mit binären Systemen<sup>6</sup> zu finden (vgl. bspw. Beichelt (1993, S. 57 ff.)), wobei die Größen  $p_{it}$  entweder Zuverlässigkeiten der Komponenten darstellen können (vgl. Anhang A.1) oder auch deren Zustände aus der Menge  $\{0; 1\}$  selbst. In letzterem Fall könnte (bei Vernachlässigung des Zeitindex) die Funktion  $\bar{\varphi}(\mathbf{p}) = \min_i p_i$  zum Beispiel eine Strukturfunktion für ein Seriensystem sein, denn falls eine Komponente defekt ist, besitzt  $\bar{\varphi}(\mathbf{p})$  den Wert 0. Analog könnte  $\bar{\varphi}(\mathbf{p}) = \max_i p_i$  eine Strukturfunktion für ein Parallelsystem sein, die den Wert 1 zurückliefert, wenn mindestens eine Komponente funktionstüchtig ist.

Falls man  $\mathbf{p} = (p_i)_{i=1, \dots, n}$  allerdings als Zuverlässigkeiten des Systems interpretiert, so ergibt sich nach Beichelt (1993, S. 57) für ein Seriensystem bspw. die Strukturfunktion

$$\bar{\varphi}(\mathbf{p}) = \prod_{i=1, \dots, n} p_i.$$

Auch bei dieser Definition ist es so, dass  $\bar{\varphi}(\mathbf{p}) = 0$  ist, falls ein  $p_i$  den Wert 0 annimmt und 1, falls für alle Zuverlässigkeiten  $p_i = 1$  gilt.

In unserem Fall von Multizustandssystemen, das als Berechnungsgrundlage für die Systemleistung prozentuale Abnutzungsvorräte und nicht Zuverlässigkeiten zugrunde legt, sind wir grundsätzlich nicht an das Konzept der Strukturfunktion gebunden. Die Anlehnung unserer Strukturbewertungsfunktion an die Strukturfunktion aus der Literatur hat allerdings u. a. folgende Vorteile: 1) Bei der Wahl von Verschleißprozessen, die sich im Gegensatz zum Gammaprozess (vgl. Abschnitt 4.2) auf Zuverlässigkeiten beziehen lassen, können die Überlegungen dieser Arbeit konzeptionell auf die Zuverlässigkeiten binärer Systeme übertragen werden, und 2) durch die vorhandene Theorie zur Strukturfunktion ist direkt ein Rüstzeug für die Kennzahlberechnung von komplexeren Systemen über eine *Pivotzerlegung* (s. Beichelt (1993, S. 63 ff.)) gegeben.

Als Mindestanforderung wollen wir für alle betrachteten Strukturbewertungsfunktionen zunächst definieren, dass diese mindestens zwei Mal ableitbar sind sowie

$$\varphi(\mathbf{0}) = 0 \text{ und } \varphi(\mathbf{1}) = 1$$

---

<sup>6</sup>Zur Erinnerung: Bei binären Systemen können die Komponenten nur die Werte 0 (Komponente defekt) und 1 (Komponente funktionstüchtig) annehmen.

gilt mit Nullvektor  $\mathbf{0}$  und einem Vektor  $\mathbf{1}$ , bei dem alle Einträge gleich Eins sind.

Im Zusammenhang mit Strukturüberlegungen stellt sich noch die Frage, wie wichtig einzelne Komponenten innerhalb eines Gesamtsystems sind. Beichelt (1993, S. 87 ff.) führt dafür den Begriff der *Importanz* im Zusammenhang mit Strukturfunktionen ein. Wir verwenden in dieser Dissertation für die Strukturbewertungsfunktion  $\varphi$  analog die Birnbaum-Importanz

$$I_i(\mathbf{p}) := \frac{\partial}{\partial p_i} \varphi(\mathbf{p}).$$

Zudem betrachten wir im weiteren Verlauf ausschließlich monotone Strukturbewertungsfunktionen, d. h., die Importanz einer Komponente ist stets nichtnegativ mit  $I_i(\mathbf{p}) \geq 0$ .

### 4.3.3 Planungszielsetzung

Die Systemleistung am Ende einer Periode  $t$ , also die Kennzahl für die Gesamtbewertung aller prozentualen Abnutzungsvorräte der Komponenten des Systems zu diesem Zeitpunkt, bezeichnen wir mit  $\phi_t$ . Die Berechnung von  $\phi_t$  erfolgt über die Strukturbewertungsfunktion, sodass

$$\phi_t := \varphi \circ \mathbf{g} : \mathbb{R}_{\geq 0}^n \longrightarrow [0; 1], \mathbf{x}_t \mapsto \phi_t(\mathbf{x}_t) = \varphi(\mathbf{g}(\mathbf{x}_t, \mathbf{p}_{t-1}, \widehat{\Delta \mathbf{w}}_t))$$

die Komposition von  $\varphi$  mit dem Vektor der Zustandsübergangsfunktionen  $\mathbf{g}$  darstellt.

Wir legen als Ziel fest, dass die minimale Systemleistung im Planungszeitraum maximiert werden soll. Die über den Planungszeitraum zu optimierende Zielsetzung der Instandhaltungsmaßnahmen wird dann über die Zielfunktion

$$f : \mathbb{R}_{\geq 0}^{n \cdot T} \longrightarrow [0; 1], \mathbf{x} \mapsto f(\mathbf{x}) := \min_{t=1, \dots, T} \phi_t(\mathbf{x}_t)$$

beschrieben.

Eine Zielfunktion, die ein Maß für die Systemleistung maximieren möchte, ist im Rahmen der Budgetallokation sinnvoll, um das zu verteilende Budget optimal auszunutzen. Bei einem endlichen Planungshorizont, wie wir ihn annehmen, würden bei einem Kostenminimierungsziel keine sinnvollen Ergebnisse erzielt. Alternative Zielfunktionen wie eine Summen- oder Mittelwertbildung der Systemleistungen über den Planungszeitraum

wären zwar denkbar, die Minimumbildung erscheint im Kontext der Instandhaltung allerdings eine natürlichere Zielstellung zu sein. Für ein gegebenes System z. B. von Straßen wird bei Minimumbildung die Instandhaltung in einer Periode nicht zugunsten der besseren Systemleistung des Straßensystems einer anderen Periode vernachlässigt. Bei einer Summen- oder Mittelwertbildung könnte dies jedoch der Fall sein.

## 4.4 Erwartungswertmodell

Wir formulieren nun das vollständige deterministische Planungsmodell und erläutern dabei noch einmal die Zielfunktion und die Nebenbedingungen im Zusammenhang, bevor wir das Modell in den Kontext der rollierenden Planung setzen. Vor dem Hintergrund der Überlegungen und Erläuterungen aus den vorangegangenen Abschnitten 4.1 bis 4.3 ergibt sich nun für das Erwartungswertmodell folgende Darstellung:

$$\text{Max.}_{\mathbf{x}} \min_{t=1, \dots, T} \phi_t \quad (4.5)$$

u. d. N.

$$\phi_t = \varphi((p_{it})_{i=1, \dots, n}) \quad (t = 1, \dots, T) \quad (4.6)$$

$$p_{it} = g_i(x_{it}, p_{i,t-1}, \widehat{\Delta w_{it}}) \quad (i = 1, \dots, n; t = 1, \dots, T) \quad (4.7)$$

$$B_t = B_{t-1} - \sum_{i=1}^n x_{it} \quad (t = 1, \dots, T) \quad (4.8)$$

$$B_t \geq 0 \quad (t = 0, \dots, T) \quad (4.9)$$

$$0 \leq p_{it} \leq 1 \quad (i = 1, \dots, n; t = 0, \dots, T) \quad (4.10)$$

$$x_{it} \geq 0 \quad (i = 1, \dots, n; t = 1, \dots, T) \quad (4.11)$$

$$B_0 = \widehat{B} \quad (4.12)$$

$$p_{i0} = \widehat{p}_i \quad (i = 1, \dots, n) \quad (4.13)$$

Die Zielfunktion (4.5) bildet die Zielsetzung ab, das Minimum der Periodensystemleistungen über die Perioden  $t = 1, \dots, T$  zu maximieren (vgl. Abschnitt 4.3.3).

Gl. (4.6) beschreibt die periodenweise Berechnung der Systemleistung über die Struk-

turbewertungsfunktion (z. B.  $\varphi = \varphi^{\text{Ser}} = \prod_{i=1}^n p_i$  für Seriensysteme oder  $\varphi = \varphi^{\text{Par}} = 1 - \prod_{i=1}^n (1 - p_i)$  für Parallelsysteme).

In Gl. (4.7) findet sich die Zustandsübergangsfunktion der Komponenten (s. Abschnitt 4.3.1) mit Periodenverschleiß  $\widehat{\Delta w}_{it}$  (vgl. Abschnitt 4.2) als Parameter. Über die Ungleichungen (4.10) wird explizit der Wertebereich der Zustandsübergangsfunktionen  $g_i$  sichergestellt. Dies ist allerdings bei entsprechender Definition der Funktionen  $g_i$  allerdings nicht notwendig (für  $t > 0$ ) und ist vor allem zur Verdeutlichung in das Modell aufgenommen worden.

Für eine rollierende Planung ist es sinnvoll, die Budgetrestriktion anstelle einer Formulierung über das Gesamtbudget (vgl. Ungl. (4.1) auf S. 36) mit einer Budgetentwicklung von Periode zu Periode darzustellen. Diese wird in Gl. (4.8) beschrieben, wobei das Restbudget  $B_t$  zum Ende von Periode  $t$  einer Nichtnegativitätsbedingung genügen muss (s. Gl. (4.9)).

Die eigentlichen Entscheidungsvariablen des Modells sind die nichtnegativen Investitionen in Instandhaltungsmaßnahmen  $x_{it}$  in Gl. (4.11). Das Gesamtbudget und die Startzustände werden über die Gleichungen (4.12) und (4.13) extern vorgegeben.

Die rollierende Planung wird mit diesem Modell wie folgt ablaufen. Es wird zunächst das oben beschriebene Erwartungswertmodell für den gesamten Planungszeitraum gelöst, tatsächlich freigegeben werden aber nur die Investitionen für die erste Periode, also  $\mathbf{x}_1 = (x_{i1})_{i=1,\dots,n}$ . Die nächste Planung findet zu Beginn von Periode  $t = 2$  mit einem Planungshorizont von  $T - 1$  Perioden statt, also für die Perioden  $t = 2, \dots, T$ . Da der Verschleiß in Periode  $t = 1$  mittlerweile realisiert wurde, gehen die prozentualen Abnutzungsvorräte  $\mathbf{p}_1 = (p_{i1})_{i=1,\dots,n}$  nun als Parameter  $\hat{\mathbf{p}} = (\hat{p}_i)_{i=1,\dots,n}$  in die zweite Periode ein. Das neue Budget für die Planung ist dann  $\hat{B}^{\text{Rest}} = B_1 = \hat{B} - \sum_{i=1}^n x_{i1}$  zu Beginn von Periode  $t = 2$ . Dieses Vorgehen wird nun für jede Periode wiederholt, bis der Planungshorizont erreicht wurde.

Im nächsten Kapitel beschäftigen wir uns mit Methoden zur Lösung des formulierten Erwartungswertmodells.

# 5 Kapitel 5

---

## Analyse des Erwartungswertmodells

Um das deterministische Erwartungswertmodell zu analysieren, werden wir insbesondere Beweise führen. Zu Beginn beschäftigen wir uns in Abschnitt 5.1 mit Vorüberlegungen, die uns zu bestimmten Eigenschaften der Lösung führen. Um zu plausibilisieren, dass eine analytische Lösung des Optimierungsproblems nur in Ausnahmefällen möglich ist, wenden wir uns einem einperiodigen Spezialfall des Problems für ein zweikomponentiges Parallelsystem in Abschnitt 5.2 zu. Abschnitt 5.3 erweitert den Blick auf ein allgemeines System, bleibt jedoch noch bei einem einperiodigen Planungszeitraum. Es wird die Bedeutung von Karush-Kuhn-Tucker-Punkten klar, die für die notwendigen Optimalitätsbedingungen erster Ordnung benötigt werden. Hier wird zudem ein einfacher Algorithmus, die Grenznutzenmethode, abgeleitet, der sich bei einem einperiodigen Planungszeitraum anwenden lässt. Abschnitt 5.4 erweitert das einperiodige Problem auf das mehrperiodige und zeigt Wege auf, wie man eine Lösung für das mehrperiodige Erwartungswertmodells erhalten kann, das die Grundlage für die rollierende Planung der Budgetallokation bildet. Wir werden zeigen, dass ein Optimierungssolver, der anhand von Karush-Kuhn-Tucker-Punkten lokal optimale Lösungen findet, genutzt werden kann, um innerhalb kurzer Rechenzeiten zumindest für Spezialfälle global optimale Lösungen zu bestimmen.

### 5.1 Vorüberlegungen

Wir wollen zeigen, dass es stets sinnvoll ist, das gesamte Budget einzusetzen, welche Investitions- bzw. Zustandsübergangsfunktion (vgl. Abschnitt 4.3.1) sinnvoll ist und was dies für die Lösungsstruktur bedeutet.

### 5.1.1 Volle Ausschöpfung des Budgets

Um zu zeigen, dass eine volle Ausschöpfung des Budgets sinnvoll ist, bei der also  $B_T = 0$  gilt, wollen wir folgenden Satz beweisen.

**Satz 1.** *Für das Erwartungswertmodell existiert stets eine global optimale Lösung mit  $B_T = 0$ .*

*Beweis.* Wir führen einen Widerspruchsbeweis. Annahme: Es existiert keine global optimale Lösung mit  $B_T = 0$ . Sei also  $B_T =: c > 0$  für jede global optimale Lösung  $\mathbf{x}'$ . Sei nun der optimale Zielfunktionswert  $z' := \phi_{t'} = \varphi(\mathbf{p}'_{t'})$  dieser optimalen Lösung, dann wird somit das maximale Minimum in Periode  $t'$  erreicht.

Wegen  $\frac{\partial}{\partial x_{it}} g_i(x_{it}, p_{i,t-1}, \widehat{\Delta w}_{it}) \geq 0$  (nach Abschnitt 4.3.1) ist für eine Komponente  $i'$ , in die wir das verbleibende Budget  $c$  investieren,

$$g_i(x'_{i't'}, p_{i',t'-1}, \widehat{\Delta w}_{i't'}) \leq g_i\left(\underbrace{x'_{i't'} + c}_{=: x''_{i't'}}, p_{i',t'-1}, \widehat{\Delta w}_{i't'}\right), \quad (5.1)$$

und wegen  $\frac{\partial}{\partial p_{it}} \varphi(\mathbf{p}_t) \geq 0$  (Monotonie der Strukturbewertungsfunktion) ist mit  $x''_{jt'} := x'_{jt'}$  für  $j \neq i'$  und  $x''_{i't'}$  für  $j = i'$  aus Gl. (5.1)

$$\varphi(\mathbf{g}(\mathbf{x}'_{t'}, \mathbf{p}'_{t'-1}, \widehat{\Delta \mathbf{w}}_{t'})) = \varphi(\mathbf{p}'_{t'}) \leq \varphi(\mathbf{p}''_{t'}) := \varphi(\mathbf{g}(\mathbf{x}''_{t'}, \mathbf{p}'_{t'-1}, \widehat{\Delta \mathbf{w}}_{t'})) =: z'',$$

wodurch  $z'' \geq z'$ . Da wir ein Maximierungsproblem betrachten, ist dies ein Widerspruch, da auch  $z''$  einen global optimalen Zielfunktionswert darstellt, in der das gesamte Budget ausgegeben wird. Somit existiert für das deterministische Erwartungswertmodell eine global optimale Lösung mit  $B_T = 0$ .  $\square$

### 5.1.2 Herleitung einer geeigneten Zustandsübergangsfunktion

Wir leiten zunächst eine konkrete Investitionsfunktion  $g_i^+(x, p)$  als Teil der Zustandsübergangsfunktion aus Abschnitt 4.3.1 her, indem wir fordern, dass eine Investition in die Wartung einer Komponente zum gleichen Komponentenzustand führen muss wie eine Sequenz kleinerer Investitionen, die in Summe den gleichen Investitionsbetrag ergeben.

Wenn keine Investition  $x$  getätigt wird, ist (bei Vernachlässigung des Komponenten- und Zeitindex)

$$g^+(0, p) = p. \quad (5.2)$$

Um die erstgenannte Forderung zu erfüllen, muss gelten:

$$g^+(x, p) \stackrel{!}{=} g^+(x - x', g^+(x', p)), \quad x \geq 0, \quad 0 < x' \leq x. \quad (5.3)$$

Die Taylor-Entwicklung erster Ordnung (vgl. Forster (2017, S. 87 ff.)) des Terms auf der rechten Seite nach beiden Dimensionen um den Punkt  $(\hat{x}, \hat{p})$  liefert

$$\begin{aligned} g^+(\hat{x} - x', g^+(x', \hat{p})) &= g^+(\hat{x}, \hat{p}) + \frac{\partial g^+}{\partial x}(\hat{x}, \hat{p}) \cdot (-x') + \frac{\partial g^+}{\partial p}(\hat{x}, \hat{p}) \cdot (g^+(x', \hat{p}) - \hat{p}) \\ &\quad + \mathcal{O}\left(\left\|(-x', (g^+(x', \hat{p}) - \hat{p}))\right\|^2\right). \end{aligned}$$

Nach Ersetzung von  $(\hat{x}, \hat{p})$  durch  $(x, p)$ , Einsetzen von Gl. (5.3) für die linke Seite, Umstellen und Division durch  $x' > 0$  erhalten wir

$$\frac{\partial g^+}{\partial x}(x, p) = \frac{\partial g^+}{\partial p}(x, p) \cdot \frac{g^+(x', p) - p}{x'} + \underbrace{\frac{\mathcal{O}\left(\|(x', (g^+(x', p) - p))\|^2\right)}{x'}}_{=0 \text{ für } x' \downarrow 0}. \quad (5.4)$$

Die Taylor-Entwicklung erster Ordnung von  $g^+(x', p)$  nach einer Dimension um den Punkt  $(0, \hat{p})$  ergibt

$$g^+(x', \hat{p}) = \underbrace{g^+(0, \hat{p})}_{\substack{\text{Gl. (5.2)} \\ = \hat{p}}} + \frac{\partial g^+}{\partial x}(0, \hat{p}) \cdot x' + \mathcal{O}(\|x'\|^2).$$

Nach Ersetzung von  $\hat{p}$  durch  $p$ , Umstellen und Division durch  $x'$  erhalten wir

$$\frac{g^+(x', p) - p}{x'} = \frac{\partial g^+}{\partial x}(0, p) + \underbrace{\frac{\mathcal{O}(\|x'\|^2)}{x'}}_{=0 \text{ für } x' \downarrow 0}. \quad (5.5)$$



Für  $x' \downarrow 0$  erhält man somit aus Gl. (5.4) und Gl. (5.5) insgesamt folgende nichtlineare partielle Differentialgleichung erster Ordnung:

$$\frac{\partial g^+}{\partial x}(x, p) = \frac{\partial g^+}{\partial p}(x, p) \cdot \frac{\partial g^+}{\partial x}(0, p), \quad x \geq 0. \quad (5.6)$$

Um eine Lösung für Gl. (5.6) zu finden, nutzen wir den Produktansatz

$$g^+(x, p) = u(x) \cdot v(p) + \beta$$

mit Anfangswertbedingung

$$g^+(0, p) = p$$

und additiver Konstante  $\beta$ , die wir am Schluss hinzufügen. Für  $p > 0$  gilt

$$\frac{\partial u(x)}{\partial x} \cdot v(p) = \frac{\partial v(p)}{\partial p} \cdot u(x) \cdot \frac{\partial u}{\partial x}(0) \cdot v(p) \quad (5.7)$$

$$\stackrel{g^+(x,p) \neq 0}{\Leftrightarrow} \quad \frac{1}{u(x)} \cdot \frac{\partial u(x)}{\partial x} = \frac{\partial v(p)}{\partial p} \cdot \frac{\partial u}{\partial x}(0) \quad (5.8)$$

$$\Leftrightarrow \quad \frac{u'(x)}{u(x)} = u'(0) \cdot v'(p) = \text{const.} =: -a. \quad (5.9)$$

Betrachte zunächst  $\frac{u'(x)}{u(x)} = -a$ :

$$\frac{u'(x)}{u(x)} = -a \quad (5.10)$$

$$\Leftrightarrow \quad \frac{du}{dx} = -a \cdot u(x) \quad (5.11)$$

$$\Leftrightarrow \quad \frac{1}{u(x)} du = -a \cdot dx \quad (5.12)$$

$$\Leftrightarrow \quad u(x) = C \cdot \exp(-ax) \quad (5.13)$$

mit einer Konstanten  $C$ . Daraus folgt  $u'(x) = -a \cdot C \cdot \exp(-ax)$ , also  $u'(0) = -a \cdot C$

Damit folgt nun aus  $u'(0) \cdot v'(p) = -a$ :

$$-a \cdot C \cdot v'(p) = -a \quad (5.14)$$

$$\Leftrightarrow v'(p) = \frac{1}{C} \quad (5.15)$$

$$\Rightarrow v(p) = \frac{p}{C} + D \quad (5.16)$$

mit einer weiteren Konstanten  $D$ . Wir fügen nun alles zusammen und erhalten

$$\begin{aligned} g^+(x, p) &= C \cdot \exp(-ax) \cdot \left( \frac{p}{C} + D \right) + \beta \\ &= p \cdot \exp(-ax) + C \cdot D \cdot \exp(-ax) + \beta. \end{aligned}$$

Die Anfangsbedingung liefert

$$g^+(0, p) = p + C \cdot D + \beta \stackrel{!}{=} p \quad (5.17)$$

$$\Leftrightarrow C \cdot D = -\beta \quad (5.18)$$

Eine Lösung der Differentialgleichung (5.6) ist also

$$g^+(x, p) = \beta - (\beta - p) \cdot \exp(-ax)$$

mit  $a, \beta \in \mathbb{R}$ . Nehmen wir nun die Anforderung der Konkavität (abnehmender Grenznutzen) und die Beschränkung  $0 \leq g^+(x, p) \leq 1$  hinzu, so erhalten wir mit einem Skalierungsparameter  $\hat{a} > 0$  und Ausgangszustand  $\hat{p}$  die Investitionsfunktion

$$g^+(x, \hat{p}) = 1 - (1 - \hat{p}) \cdot \exp(-\hat{a}x).$$

In Anhang A.2 auf Seite 148 wird verifiziert, dass diese Gleichung nicht nur für  $x' \downarrow 0$  gilt sondern auch für  $0 < x' \leq x$ .

Mit dieser Investitionsfunktion ergibt sich als Ableitung die Effektivitätsfunktion der

Investition  $\eta(x) = \hat{a} \cdot (1 - \hat{p}) \cdot \exp(-\hat{a}x)$ . Sie beschreibt den Zuwachs des Grenznutzens von  $x$  bzw. die Wirkung der Investition einer zusätzlichen Geldeinheit (GE) auf den prozentualen Abnutzungsvorrat einer Komponente.

Mit der soeben entwickelten Investitionsfunktion folgt durch Subtraktion des erwarteten Periodenverschleißes und Maximumsbildung mit 0 die folgende Zustandsübergangsfunktion:

$$g_i(x_{it}, p_{i,t-1}, \widehat{\Delta w}_{it}) = p_{it} = \max \left\{ 1 - (1 - p_{i,t-1}) \cdot \exp(-\hat{a}_i x_{it}) - \widehat{\Delta w}_{it}; 0 \right\}.$$

### 5.1.3 Eigenschaften der Lösung der gewählten Zustandsübergangsfunktion

Wir werden nun zeigen, dass die Folge der Periodensystemleistungen des deterministischen Erwartungswertmodells für diese Funktion des Zustandsübergangs eine besondere Struktur besitzt. Wir gehen zur Vereinfachung in diesem Abschnitt davon aus, dass stets für alle Komponenten  $i$  und alle Perioden  $t$

$$1 - (1 - p_{i,t-1}) \cdot \exp(-\hat{a}_i \cdot x_{it}) \geq \widehat{\Delta w}_{it}$$

gilt. Als Vorbereitung für die weitere Untersuchung benötigen wir folgendes Lemma.

**Lemma 2.** *Sei  $t' < T$  und  $\Delta x_i > 0$  ein zusätzlicher Einsatz von Budget, der entweder in einer Periode  $t'$  oder Periode  $t' + 1$  in Komponente  $i$  investiert wird. Dann gilt für  $\widehat{\Delta w}_{it} > 0$ :*

$$\begin{aligned} p'' &:= g_i(x_{i,t'+1} + \Delta x_i, g_i(x_{it'}, p_{i,t'-1}, \widehat{\Delta w}_{it'}), \widehat{\Delta w}_{i,t'+1}) \\ &> g_i(x_{i,t'+1}, g_i(x_{it'} + \Delta x_i, p_{i,t'-1}, \widehat{\Delta w}_{it'}), \widehat{\Delta w}_{i,t'+1}) =: p'. \end{aligned}$$

*Beweis.* Betrachte zunächst die Periode  $t'$  und definiere

$$\begin{aligned} \bar{p} &:= 1 - (1 - p_{i,t'-1}) \cdot \exp(-\hat{a}_i \cdot (x_{it'} + \Delta x_i)) - \widehat{\Delta w}_{it'}, \\ \bar{\bar{p}} &:= 1 - (1 - p_{i,t'-1}) \cdot \exp(-\hat{a}_i \cdot x_{it'}) - \widehat{\Delta w}_{it'}. \end{aligned}$$

Für die Periode  $t' + 1$  gilt dann:

$$\begin{aligned} p' &= 1 - (1 - \bar{p}) \cdot \exp(-\hat{a}_i \cdot x_{i,t'+1}) - \widehat{\Delta w}_{i,t'+1}, \\ p'' &= 1 - (1 - \bar{\bar{p}}) \cdot \exp(-\hat{a}_i \cdot (x_{i,t'+1} + \Delta x_i)) - \widehat{\Delta w}_{i,t'+1}. \end{aligned}$$

Berechne die Differenz und setze ein:

$$\begin{aligned} p'' - p' &= - (1 - \bar{\bar{p}}) \cdot \exp(-\hat{a}_i \cdot (x_{i,t'+1} + \Delta x_i)) \\ &\quad + (1 - \bar{p}) \cdot \exp(-\hat{a}_i \cdot x_{i,t'+1}) \\ &= - (1 - \bar{\bar{p}}) \cdot \exp(-\hat{a}_i \cdot x_{i,t'+1}) \cdot \exp(-\hat{a}_i \cdot \Delta x_i) \\ &\quad + (1 - \bar{p}) \cdot \exp(-\hat{a}_i \cdot x_{i,t'+1}) \\ &= \exp(-\hat{a}_i \cdot x_{i,t'+1}) \cdot \left( - (1 - \bar{\bar{p}}) \cdot \exp(-\hat{a}_i \cdot \Delta x_i) + (1 - \bar{p}) \right). \end{aligned}$$

Wegen  $\exp(-\hat{a}_i \cdot x_{i,t'+1}) > 0$  kann durch diesen Term dividiert werden und man erhält

$$\begin{aligned} &- (1 - \bar{\bar{p}}) \cdot \exp(-\hat{a}_i \cdot \Delta x_i) + 1 - \bar{p} \\ &= - \left( 1 - (1 - (1 - p_{i,t'-1}) \cdot \exp(-\hat{a}_i \cdot x_{it'}) - \widehat{\Delta w}_{it'}) \right) \cdot \exp(-\hat{a}_i \cdot \Delta x_i) \\ &\quad + 1 - \left( 1 - (1 - p_{i,t'-1}) \cdot \exp(-\hat{a}_i \cdot (x_{it'} + \Delta x_i)) - \widehat{\Delta w}_{it'} \right) \\ &= - \left( (1 - p_{i,t'-1}) \cdot \exp(-\hat{a}_i \cdot x_{it'}) + \widehat{\Delta w}_{it'} \right) \cdot \exp(-\hat{a}_i \cdot \Delta x_i) \\ &\quad + (1 - p_{i,t'-1}) \cdot \exp(-\hat{a}_i \cdot (x_{it'} + \Delta x_i)) + \widehat{\Delta w}_{it'} \\ &= - (1 - p_{i,t'-1}) \cdot \exp(-\hat{a}_i \cdot (x_{it'} + \Delta x_i)) - \widehat{\Delta w}_{it'} \cdot \exp(-\hat{a}_i \cdot \Delta x_i) \\ &\quad + (1 - p_{i,t'-1}) \cdot \exp(-\hat{a}_i \cdot (x_{it'} + \Delta x_i)) + \widehat{\Delta w}_{it'} \\ &= \widehat{\Delta w}_{it'} \cdot (1 - \exp(-\hat{a}_i \cdot \Delta x_i)) > 0. \end{aligned}$$

□

Lemma 2 sagt aus, dass eine zusätzliche Investition generell zu einem späteren Zeitpunkt eine größere Wirkung besitzt. Hiermit können wir nun folgenden Satz beweisen.

**Satz 3.** Sei  $\phi_t = \varphi(\mathbf{p}_t)$  die Perioden-Systemleistung (s. Gl. (4.6)) des deterministischen Erwartungswertmodells mit Zustandsübergangsfunktion

$$p_{it} = g_i(x_{it}, p_{i,t-1}, \widehat{\Delta w}_{it}) = 1 - (1 - p_{i,t-1}) \cdot \exp(-\hat{a}_i \cdot x_{it}) - \widehat{\Delta w}_{it}.$$

Dann existiert eine optimale Lösung  $\mathbf{x}^*$ , bei der  $\phi_1^* \geq \dots \geq \phi_T^*$  mit optimalem Zielfunktionswert  $z^* = \phi_T^*$ .

*Beweis.* Wir führen erneut einen Widerspruchsbeweis und nehmen an, dass  $\mathbf{x}'$  die optimale Lösung mit Zielfunktionswert  $z'$  sei und diese Struktur der Lösung nicht gelte. Dann gilt für mindestens ein  $t' < T$  in dieser optimalen Lösung, dass  $\phi_{t'} < \phi_{t'+1}$ . Sei nun  $\bar{x}_t := \sum_{i=1, \dots, n} x_{it}$  die Gesamtinvestition in einer Periode  $t$ . Nehmen wir zunächst an, dass  $t' = T - 1$ , also  $\phi_{T-1}' > \phi_T'$ . Wegen der Ableitbarkeit ist  $\varphi$  insbesondere stetig und somit der Zwischenwertsatz (vgl. Forster (2016, S. 114 ff.)) anwendbar. Hiernach gibt es ein  $\bar{x}_T'' < \bar{x}_T'$ , sodass  $\phi_{T-1}' = \phi_T''$ . Für den Fall, dass  $z^* = \phi_{T-1}'$  haben wir also schon ein zulässiges Gegenbeispiel gefunden, denn damit hat man die Struktur der nicht steigenden Perioden-Systemleistungen erhalten und  $z^* = \phi_{T-1}' = \phi_T''$ .

Betrachten wir nun den Fall  $t' < T - 1$  mit  $\phi_{t'+1}' > \phi_{t'}'$ . Dann gibt es ein  $\mathbf{x}_{t'+1}'' < \mathbf{x}_{t'+1}'$ , sodass  $\phi_{t'}' = \phi_{t'+1}''$  (dazu vermindere man  $\mathbf{x}_{t'+1}'$  iterativ komponentenweise für alle  $i$ , für die  $x_{i,t'+1}' > 0$ , bis  $x_{i,t'+1}' = 0$  oder der Zwischenwertsatz anwendbar ist).<sup>1</sup> Sei nun  $\Delta \mathbf{x} = \mathbf{x}_{t'+1}' - \mathbf{x}_{t'+1}''$  der Differenzvektor. Wir wenden nun Lemma 2 auf jede einzelne Komponente  $i$  an, für die  $\Delta x_i > 0$  gilt, sodass  $\mathbf{x}_{t'+2}'' = \mathbf{x}_{t'+2}' + \Delta \mathbf{x}$ , verschieben also das freigewordene Budget auf einen späteren Zeitpunkt bei jeder Komponente  $i$  mit  $\Delta x_i > 0$ . Wegen der Monotonie der Strukturbewertungsfunktion  $\frac{\partial}{\partial p_i} \varphi(\mathbf{p}) \geq 0$  (vgl. Abschnitt 4.3.2) gilt weiterhin  $z'' \geq z'$ . Dieses Vorgehen kann man solange iterativ anwenden, bis die gewünschte Struktur der nicht steigenden Perioden-Systemleistungen hergestellt ist. Dieser Widerspruch schließt den Beweis.  $\square$

Satz 3 bedeutet im Wesentlichen, dass in einer optimalen Lösung die Perioden-Systemleistungen im Zeitverlauf nichtwachsend sind und daher die minimale Perioden-Systemleistung und damit der optimale Zielfunktionswert am Ende der letzten Periode des Planungszeitraums angenommen wird.

<sup>1</sup>Das Ungleichheitszeichen zwischen Vektoren ist komponentenweise zu verstehen, wobei es genügt, dass für eine Komponente  $i$  die strikte Ungleichung „<“ gilt und für die übrigen „≤“.

## 5.2 Analytische Lösung des Erwartungswertmodells für Zwei-Komponenten-Parallelsysteme bei einperiodigem Planungszeitraum

Um uns einer Lösungsmethodik für das mehrperiodige Erwartungswertmodell zu nähern, lösen wir zunächst analytisch einen exemplarischen Spezialfall<sup>2</sup> des Problems. Wir betrachten eines der einfachsten modularen Systeme, nämlich ein Parallelsystem mit nur zwei Komponenten, für nur eine Periode. Zur Vereinfachung werden wir in diesem Abschnitt den Zeitindex vernachlässigen. Da eine optimale Lösung existiert, in der das Gesamtbudget voll ausgeschöpft wird (vgl. Abschnitt 5.1.1), eliminieren wir die Entscheidungsvariable  $x_2$ , indem wir sie durch  $x_2 = \hat{B} - x_1$  ersetzen. Für die Zustandsübergangsfunktion nehmen wir erneut

$$g_i(x_i, \hat{p}_i, \widehat{\Delta w}_i) = 1 - (1 - \hat{p}_i) \cdot \exp(-\hat{a}_i \cdot x_i) - \widehat{\Delta w}_i \geq 0$$

an. Als Strukturbewertungsfunktion definieren wir hier in Anlehnung an Beichelt (1993, S. 57 ff.)

$$\varphi(\mathbf{p}) = 1 - \prod_{i=1, \dots, n} (1 - p_i).$$

Das Optimierungsproblem für ein zweikomponentiges System für nur eine Periode lautet somit

$$\text{Max.}_{\mathbf{x}} \quad \varphi(p_1, p_2) = 1 - (1 - p_1) \cdot (1 - p_2)$$

u. d. N.

$$\begin{aligned} p_1 &= g_1(x_1, \hat{p}_1, \widehat{\Delta w}_1) \\ &= 1 - (1 - \hat{p}_1) \cdot \exp(-\hat{a}_1 \cdot x_1) - \widehat{\Delta w}_1 \\ p_2 &= g_2(\hat{B} - x_1, \hat{p}_2, \widehat{\Delta w}_2) \\ &= 1 - (1 - \hat{p}_2) \cdot \exp(-\hat{a}_2 \cdot (\hat{B} - x_1)) - \widehat{\Delta w}_2 \\ 0 &\leq x_1 \leq \hat{B}. \end{aligned}$$

Damit können wir das Optimierungsproblem wie folgt als ein Maximierungsproblem in einer Variablen formulieren (die Bedingung  $x_1 \geq 0$  fügen wir erst später wieder hinzu).

<sup>2</sup>Eine analoge Vorgehensweise für ein zweikomponentiges Seriensystem findet sich in Anhang A.3.2.

$$\begin{aligned}
 & \text{Max.}_{x_1} \quad 1 - \left(1 - \left(1 - (1 - \hat{p}_1) \cdot \exp(-\hat{a}_1 x_1) - \widehat{\Delta w}_1\right)\right) \\
 & \quad \cdot \left(1 - \left(1 - (1 - \hat{p}_2) \cdot \exp(-\hat{a}_2 \cdot (\hat{B} - x_1)) - \widehat{\Delta w}_2\right)\right) \\
 \Leftrightarrow & \text{Min.}_{x_1} \quad f^{\min}(x_1) := \left((1 - \hat{p}_1) \cdot \exp(-\hat{a}_1 x_1) + \widehat{\Delta w}_1\right) \\
 & \quad \cdot \left((1 - \hat{p}_2) \cdot \exp(-\hat{a}_2 \cdot (\hat{B} - x_1)) + \widehat{\Delta w}_2\right)
 \end{aligned}$$

Berechne nun die Ableitung  $\frac{\partial}{\partial x_1} f^{\min}(x_1)$  und setze diese gleich 0, um Kandidaten für lokale Optima zu bestimmen:

$$\begin{aligned}
 & \frac{\partial}{\partial x_1} f^{\min}(x_1) \\
 = & \frac{\partial}{\partial x_1} \left( (1 - \hat{p}_1) \cdot (1 - \hat{p}_2) \cdot \exp(-\hat{a}_1 x_1 - \hat{a}_2 \hat{B} + \hat{a}_2 x_1) \right. \\
 & \quad + \widehat{\Delta w}_1 \cdot (1 - \hat{p}_2) \cdot \exp(-\hat{a}_2 \hat{B} + \hat{a}_2 x_1) \\
 & \quad \left. + \widehat{\Delta w}_2 \cdot (1 - \hat{p}_1) \cdot \exp(-\hat{a}_1 x_1) + \widehat{\Delta w}_1 \widehat{\Delta w}_2 \right).
 \end{aligned}$$

Da die Konstanten beim Ableiten verschwinden, kann die Formel vereinfacht werden zu

$$\begin{aligned}
 = & \frac{\partial}{\partial x_1} \left( (1 - \hat{p}_1) \cdot (1 - \hat{p}_2) \cdot \exp(-\hat{a}_1 x_1 - \hat{a}_2 \hat{B} + \hat{a}_2 x_1) \right. \\
 & \quad \left. + \widehat{\Delta w}_1 \cdot (1 - \hat{p}_2) \cdot \exp(-\hat{a}_2 \hat{B} + \hat{a}_2 x_1) + \widehat{\Delta w}_2 \cdot (1 - \hat{p}_1) \cdot \exp(-\hat{a}_1 x_1) \right) \\
 = & (1 - \hat{p}_1) \cdot (1 - \hat{p}_2) \cdot \exp(-\hat{a}_2 \hat{B}) \cdot (\hat{a}_2 - \hat{a}_1) \cdot \exp((\hat{a}_2 - \hat{a}_1) x_1) \\
 & \quad + \widehat{\Delta w}_1 \cdot (1 - \hat{p}_2) \cdot \exp(-\hat{a}_2 \hat{B}) \cdot \hat{a}_2 \cdot \exp(\hat{a}_2 x_1) \\
 & \quad - \widehat{\Delta w}_2 \cdot (1 - \hat{p}_1) \cdot \hat{a}_1 \cdot \exp(-\hat{a}_1 x_1) \stackrel{!}{=} 0.
 \end{aligned}$$

Die zweite Ableitung findet sich in Anhang A.3.1.

Für die analytische Berechnung der Kandidaten für lokale Optima transformieren wir die Optimierungsvariable, indem wir  $y := \exp(x_1)$  setzen, sortieren etwas um und erhalten

$$(1 - \hat{p}_1) \cdot (1 - \hat{p}_2) \cdot \exp(-\hat{a}_2 \hat{B}) \cdot (\hat{a}_2 - \hat{a}_1) \cdot y^{\hat{a}_2 - \hat{a}_1}$$

$$\begin{aligned}
 & + \widehat{\Delta w}_1 \cdot (1 - \hat{p}_2) \cdot \exp(-\hat{a}_2 \hat{B}) \cdot \hat{a}_2 \cdot y^{\hat{a}_2} \\
 & - \widehat{\Delta w}_2 \cdot (1 - \hat{p}_1) \cdot \hat{a}_1 \cdot y^{-\hat{a}_1} \stackrel{!}{=} 0.
 \end{aligned}$$

Wir multiplizieren die Gleichung mit  $y^{\hat{a}_1} > 0$  und setzen der Übersichtlichkeit halber  $\hat{a}_1 := m \cdot \hat{a}_2$  mit  $m > 0$ , da  $\hat{a}_1, \hat{a}_2, y^{\hat{a}_1} > 0$ . Dies führt uns zur Gleichung

$$\begin{aligned}
 & (1 - \hat{p}_1) \cdot (1 - \hat{p}_2) \cdot \exp(-\hat{a}_2 \hat{B}) \cdot ((1 - m) \cdot \hat{a}_2) \cdot y^{\hat{a}_2} \\
 & + \widehat{\Delta w}_1 \cdot (1 - \hat{p}_2) \cdot \exp(-\hat{a}_2 \hat{B}) \cdot \hat{a}_2 \cdot y^{(1+m) \cdot \hat{a}_2} \\
 & - \widehat{\Delta w}_2 \cdot (1 - \hat{p}_1) \cdot m \cdot \hat{a}_2 \stackrel{!}{=} 0,
 \end{aligned}$$

die wir durch Einführung von  $z := y^{\hat{a}_2}$  umwandeln in

$$\begin{aligned}
 & \underbrace{\widehat{\Delta w}_1 \cdot (1 - \hat{p}_2) \cdot \exp(-\hat{a}_2 \hat{B}) \cdot \hat{a}_2 \cdot z^{m+1}}_{=: c_{m+1}} \\
 & + \underbrace{(1 - \hat{p}_1) \cdot (1 - \hat{p}_2) \cdot \exp(-\hat{a}_2 \hat{B}) \cdot ((1 - m) \cdot \hat{a}_2) \cdot z}_{=: c_1(m)} \\
 & \quad - \underbrace{\widehat{\Delta w}_2 \cdot (1 - \hat{p}_1) \cdot m \cdot \hat{a}_2}_{=: c_0(m)} \\
 & \stackrel{!}{=} 0,
 \end{aligned}$$

also

$$c_{m+1} \cdot z^{m+1} + c_1(m) \cdot z + c_0(m) \stackrel{!}{=} 0. \quad (5.19)$$

Falls  $m$  ganzzahlig ist, haben wir hier ein Polynom vom Grad  $(m + 1)$  mit Koeffizienten der Terme  $j$ -ter Ordnung  $c_j$ .



### 5.2.1 Analyse für ausgewählte Verhältnisse der Skalierungsparameter der Investitionsfunktion

Wir analysieren das Polynom nun für verschiedene Werte von  $m = \frac{\hat{a}_1}{\hat{a}_2}$ , also dem Verhältnis der Skalierungsparameter der Investitionsfunktion (vgl. Abschnitt 5.1.2).

a)  $m = \frac{\hat{a}_1}{\hat{a}_2} = 1$

Es verbleibt für  $m = 1$  der Term

$$\underbrace{\widehat{\Delta w}_1 \cdot (1 - \hat{p}_2) \cdot \exp(-\hat{a}_2 \hat{B}) \cdot \hat{a}_2 \cdot z^2}_{=:c_2 > 0} - \underbrace{\widehat{\Delta w}_2 \cdot (1 - \hat{p}_1) \cdot \hat{a}_2}_{=:c_0 < 0} \stackrel{!}{=} 0.$$

Durch Anwendung der „ $p$ - $q$ -Formel“ ergibt sich wegen  $c_2 > 0$  und  $c_0 < 0$  als Lösung

$$z = \sqrt{\frac{-c_0}{c_2}} > 0. \quad (5.20)$$

Somit kann über die Beziehung  $z = y^{\hat{a}_2} = \exp(\hat{a}_2 \cdot x_1)$  der Wert  $x_1 = \frac{\ln z}{\hat{a}_2}$  als Kandidat für ein lokales Optimum bestimmt werden. Aufgrund der Nebenbedingungen  $0 \leq x_1 \leq \hat{B}$  ergeben sich  $x_1 = 0$  und  $x_1 = \hat{B}$  als weitere Kandidaten. Den Wert für  $x_2$  erhält man über  $x_2 = \hat{B} - x_1$ .

b)  $m = \frac{\hat{a}_1}{\hat{a}_2} = 2$

Es verbleibt für  $m = 2$  der Term

$$\underbrace{\widehat{\Delta w}_1 \cdot (1 - \hat{p}_2) \cdot \exp(-\hat{a}_2 \hat{B}) \cdot \hat{a}_2 \cdot z^3}_{=:c_3 > 0} + \underbrace{(1 - \hat{p}_1) \cdot (1 - \hat{p}_2) \cdot \exp(-\hat{a}_2 \hat{B}) \cdot (-\hat{a}_2) \cdot z}_{=:c_1 < 0} - \underbrace{\widehat{\Delta w}_2 \cdot (1 - \hat{p}_1) \cdot 2 \cdot \hat{a}_2}_{=:c_0 < 0}$$

$$\stackrel{!}{=} 0$$

Die Diskriminante  $D_3$  für kubische Polynome berechnet sich mit Koeffizienten der Terme  $j$ -ter Ordnung  $c_j$  durch

$$\begin{aligned} D_3 &= c_2^2 c_1^2 - 4c_3 c_1^3 - 4c_2^3 c_0 + 18c_3 c_2 c_1 c_0 - 27c_3^2 c_0^2 \\ &= -4c_3 c_1^3 - 27c_3^2 c_0^2, \end{aligned}$$

denn  $c_3 > 0, c_2 = 0$  und  $c_1 < 0$ . Unter Verwendung von  $D_3$  kann eine analytische Berechnung der Lösungen mit Hilfe der Cardanischen Formeln erfolgen.

c)  $\mathbf{m} = \frac{\hat{\mathbf{a}}_1}{\hat{\mathbf{a}}_2} = \mathbf{3}$

Es verbleibt für  $m = 3$  der Term

$$\begin{aligned} &\underbrace{\widehat{\Delta w}_1 \cdot (1 - \hat{p}_2) \cdot \exp(-\hat{a}_2 \hat{B}) \cdot \hat{a}_2 \cdot z^4}_{=: c_4 > 0} \\ &+ \underbrace{(1 - \hat{p}_1) \cdot (1 - \hat{p}_2) \cdot \exp(-\hat{a}_2 \hat{B}) \cdot (-2 \cdot \hat{a}_2) \cdot z}_{=: c_1 < 0} \\ &\quad \underbrace{-\widehat{\Delta w}_2 \cdot (1 - \hat{p}_1) \cdot 3 \cdot \hat{a}_2}_{=: c_0 < 0} \\ &\stackrel{!}{=} 0 \end{aligned}$$

Die Diskriminante  $D_4$  für quartische Polynome berechnet sich mit Koeffizienten der Terme  $j$ -ter Ordnung  $c_j$  durch (Terme mit  $c_j = 0$  wurden schon entfernt):

$$D_4 = 256c_4^3 c_0^3 - 27c_4^2 c_1^4 < 0,$$

denn  $c_4 > 0, c_2 = c_3 = 0$  und  $c_0 < 0$ , d. h., mindestens eine der Nullstellen ist komplex. Auch für diesen Fall ist eine analytische Lösung noch möglich.

### 5.2.2 Bewertung

Für andere Werte von  $m$  ist eine analytische Berechnung i. d. R. nicht möglich. Schon für das relativ einfache zweikomponentige Parallelsystem mit einem Planungszeitraum von nur einer Periode hängt die Bestimmung einer analytischen Lösung stark von den vorliegenden Problemparametern ab: Ob eine analytische Lösung möglich ist, bestimmt das Verhältnis  $m = \frac{\hat{a}_1}{a_2}$  der Skalierungsparameter der Investitionsfunktion. Eine effiziente numerische Lösung kann häufig dennoch erreicht werden, wie wir im weiteren Verlauf dieser Arbeit zeigen wollen.

## 5.3 Ableitung einer Grenznutzenmethode für das einperiodige Erwartungswertmodell

Da die analytische Lösung schon für kleine Probleme schnell an seine Grenzen stößt, wollen wir zunächst ein numerisches Verfahren für das einperiodige Problem entwickeln, das auch bei komplexeren Strukturbewertungsfunktionen und einer größeren Anzahl von Komponenten angewendet werden kann.

Wir betrachten nun ein beliebiges System mit monotoner Strukturbewertungsfunktion  $\varphi$  und einem Planungshorizont von  $T = 1$ , sodass der Zeitindex erneut vernachlässigbar ist. Sei

$$\mathbf{p} := \mathbf{g}(\mathbf{x}, \hat{\mathbf{p}}, \widehat{\Delta \mathbf{w}}) = \left( g_i(x_i, \hat{p}_i, \widehat{\Delta w}_i) \right)_{i=1, \dots, n}. \quad (5.21)$$

Das deterministische Erwartungswertmodell aus Abschnitt 4.4 vereinfacht sich zu

$$\begin{aligned} & \text{Max.}_{\mathbf{x}} \varphi(\mathbf{g}(\mathbf{x}, \hat{\mathbf{p}}, \widehat{\Delta \mathbf{w}})) \\ & \text{u. d. N.} \\ & \hat{B} = \sum_{i=1}^n x_i \end{aligned} \quad (5.22)$$

$$x_i \geq 0 \quad (i = 1, \dots, n) \quad (5.23)$$

Die Gleichheit in Gl. (5.22) können wir wegen des weiter oben bewiesenen Satzes 1 auf Seite 46 annehmen.

### 5.3.1 Notwendige Optimalitätsbedingungen erster Ordnung

Grundlage der folgenden Analyse, um allgemeine Informationen über die Form der Lösung zu erhalten, ist unter anderem die Auswertung der sog. Karush-Kuhn-Tucker-Bedingungen („KKT-Bedingungen“, vgl. Bazaraa u. a. (2006, S. 165 ff.)) als Standard-Methode für die Untersuchung notwendiger Optimalitätsbedingungen. Die Lagrangefunktion  $L$ , die für das zugehörige *Minimierungsproblem*

$$\text{Min.}_{\mathbf{x}} \quad -\varphi(\mathbf{g}(\mathbf{x}, \hat{\mathbf{p}}, \widehat{\Delta \mathbf{w}})) \quad (5.24)$$

mit Nebenbedingungen (5.22) und (5.23) definiert ist, lautet mit Lagrangemultiplikator  $\lambda^B$

$$L(\mathbf{x}, \lambda^B) = -\varphi(\mathbf{g}(\mathbf{x}, \hat{\mathbf{p}}, \widehat{\Delta \mathbf{w}})) + \lambda^B \cdot \left( \hat{B} - \sum_{i=1}^n x_i \right).$$

Daraus leiten sich die KKT-Bedingungen ab. In einem KKT-Punkt  $(\mathbf{x}, \lambda^B)$  muss folgendes gelten:

$$-\frac{\partial}{\partial x_i} \varphi(\mathbf{g}(\mathbf{x}, \hat{\mathbf{p}}, \widehat{\Delta \mathbf{w}})) + \lambda^B \geq 0 \quad (i = 1, \dots, n) \quad (5.25)$$

$$\hat{B} - \sum_{i=1}^n x_i = 0$$

$$x_i \cdot \left( -\frac{\partial}{\partial x_i} \varphi(\mathbf{g}(\mathbf{x}, \hat{\mathbf{p}}, \widehat{\Delta \mathbf{w}})) + \lambda^B \right) = 0 \quad (i = 1, \dots, n) \quad (5.26)$$

$$x_i \geq 0 \quad (i = 1, \dots, n)$$

Die Linear Independence Constraint Qualification (LICQ)<sup>3</sup> ist trivialerweise erfüllt, da die Gradienten der Restriktionsfunktionen zu den Nichtnegativitätsbedingungen den kanonischen Einheitsvektoren entsprechen und wegen  $\hat{B} > 0$  nicht alle Nichtnegativitätsbedingungen bindend sein können. Wir müssen allerdings für die Anwendung der KKT-Bedingungen voraussetzen, dass für alle Komponenten  $i$  der Periodenverschleiß

<sup>3</sup>Für die Anwendung der KKT-Bedingungen muss mindestens eine von verschiedenen Constraint Qualifications zutreffen (vgl. Bazaraa u. a. (2006, S. 165 ff.)).

klein genug ist, sodass

$$\widehat{p}_i > \widehat{\Delta w}_i$$

gilt, damit die Zielfunktion als stetig differenzierbar angenommen werden kann.

Für den Grenznutzen der Investition in Komponente  $i$  gilt dann<sup>4</sup>

$$\begin{aligned} \frac{\partial}{\partial x_i} \varphi(\mathbf{g}(\mathbf{x}, \widehat{\mathbf{p}}, \widehat{\Delta \mathbf{w}})) &\stackrel{\text{Gl. (5.21)}}{=} \frac{\partial}{\partial p_i} \varphi(\mathbf{p} = \mathbf{g}(\mathbf{x}, \widehat{\mathbf{p}}, \widehat{\Delta \mathbf{w}})) \cdot \frac{\partial}{\partial x_i} g_i(x_i, \widehat{p}_i, \widehat{\Delta w}_i) \\ &= \underbrace{I_i(\mathbf{g}(\mathbf{x}, \widehat{\mathbf{p}}, \widehat{\Delta \mathbf{w}}))}_{\text{Importanz}} \cdot \underbrace{\eta_i(x_i)}_{\text{Effektivität}} \geq 0. \end{aligned}$$

Für den Fall  $x_i > 0$  folgt aus der Komplementaritätsbedingung (5.26), dass

$$\frac{\partial}{\partial x_i} \varphi(\mathbf{g}(\mathbf{x}, \widehat{\mathbf{p}}, \widehat{\Delta \mathbf{w}})) = \lambda^B, \quad (5.27)$$

und für  $x_i = 0$  aus Gl. (5.25), dass

$$\frac{\partial}{\partial x_i} \varphi(\mathbf{g}(\mathbf{x}, \widehat{\mathbf{p}}, \widehat{\Delta \mathbf{w}})) \leq \lambda^B. \quad (5.28)$$

Aus diesen Überlegungen können wir zwei Dinge schlussfolgern, die wir bei der Konstruktion der im folgenden Abschnitt behandelten Grenznutzenmethode ausnutzen wollen:

1. Der Grenznutzen aller Komponenten  $i$  mit  $x_i > 0$  ist identisch, s. Gl. (5.27).
2. Der Grenznutzen aller Komponenten  $i$  mit  $x_i > 0$  ist maximal, s. Gl. (5.28).

Die Grenznutzen der einzelnen Komponenten  $i$  beeinflussen sich also gegenseitig und „pendeln“ sich in einem Kandidaten für ein lokales Optimum bei einem Wert  $\lambda^B$  „ein“, wenn wir ein primales Gradientenverfahren in Richtung der Komponente mit dem jeweils höchsten Grenznutzen und mit abnehmender Schrittweite nutzen. Weiterführende Überlegungen zu den hinreichenden Optimalitätsbedingungen zweiter Ordnung finden sich im Anhang A.3.3.

---

<sup>4</sup>Die Begriffe Effektivität und Importanz wurden in den Abschnitten 4.3.1 und 4.3.2 eingeführt.

### 5.3.2 Grenznutzenmethode

Bei der Grenznutzenmethode aus Alg. 5.1 handelt es sich um ein einfaches primales Gradientenverfahren, d. h., nach der Initialisierung bleibt in jeder Iteration die bis dahin gefundene Lösung zulässig. Ein Optimierungsschritt erfolgt jeweils für eine Komponente  $i \in \{1, \dots, n\}$  mit einer Schrittweite  $v$ , die gegen 0 konvergiert. Wir bezeichnen dafür den Grenznutzen einer Komponente für die jeweils aktuell geltende Lösung mit  $\gamma_i$ ,  $C$  sei die Menge aller Komponenten,  $L^+$  die Teilmenge der „aussichtsreichen“ Komponenten und  $L^-$  entsprechend die Menge der „weniger aussichtsreichen“. Die Menge  $C'$  stellt in einer Iterationsschleife die noch zu untersuchende Teilmenge von  $C$  dar.

---

**Algorithmus 5.1** Grenznutzenmethode für das einperiodige Erwartungswertmodell

---

Quelle: Eigene Darstellung

---

- 0: **Initialisierung:** Setze  $(x_i)_{i=1, \dots, n} = \mathbf{x} := \mathbf{0}$  und seien die Grenznutzen  $\gamma_i^0 := \frac{\partial}{\partial x_i} \varphi(\mathbf{g}(\mathbf{0}, \hat{\mathbf{p}}, \widehat{\Delta \mathbf{w}}))$  gegeben. Setze  $C := \{1, \dots, n\}$ ,  $L^+ := \arg \max_{i \in C} \gamma_i^0$ .  $L^- := \emptyset$ . Setze den aufzuteilenden Betrag  $v := \hat{B}$  und  $\mu := 1$ . Für ein (z. B. mit minimalem Index)  $i' \in L^+$  setze  $x_{i'} \leftarrow x_{i'} + v$ .
- 1: **while** (Abbruchbedingung nicht erfüllt):
- 2:      $C' := C, L^- \leftarrow \arg \min_{i \in C'} \gamma_i$  mit  $\gamma_i := I_i(\mathbf{g}(\mathbf{x}, \hat{\mathbf{p}}, \widehat{\Delta \mathbf{w}})) \cdot \eta_i(x_i)$
- 3:     **while** (es existiert  $i \in L^-$  mit  $x_i = 0$ ):
- 4:         Aktualisiere  $C' \leftarrow C' \setminus \{i\}$  und  $L^- \leftarrow \arg \min_{i \in C'} \gamma_i$
- 5:     **if**  $L^- \subset L^+$  oder  $L^+ \subset L^-$ :  $\mu \leftarrow \mu + 1, v \leftarrow \frac{\hat{B}}{\mu}$
- 6:     Für ein  $i' \in L^-$ :  $x_{i'} \leftarrow x_{i'} - v$
- 7:     Update  $L^+ \leftarrow \arg \max_{i \in C'} \gamma_i$
- 8:     Für ein  $i' \in L^+$ :  $x_{i'} \leftarrow x_{i'} + v$
- 

Bei der Initialisierung werden die Grenznutzen aller Komponenten berechnet, wenn noch kein Budget investiert wurde. Diejenigen Komponentenindizes mit maximalem Grenznutzen (dies kann auch nur ein einziges Element sein) werden in  $L^+$  gespeichert,  $L^-$  bleibt am Anfang leer. Die Schrittweite  $v$  wird mit dem Gesamtbudget initialisiert und ein Zähler  $\mu$  auf 1 gesetzt. Es wird eine (vorläufige) Investition in eine ausgewählte Komponente  $i'$  aus der Menge der aussichtsreichen Komponenten in Höhe des Betrags  $v = \hat{B}$  geplant. Dies ist eine zulässige Startlösung für den Algorithmus, mit der sich gleichzeitig die Grenznutzen aller Komponenten durch die (vorläufige) Investition neu berechnen. Insbesondere verringert sich durch das Gesetz des abnehmenden Grenznutzens der Grenznutzen von  $i'$ .

Die Grenznutzenmethode verfolgt nun das Ziel, in jeder Iteration geschickt das Budget hinsichtlich eines besseren Zielfunktionswertes zwischen den Komponenten zu verschieben, und zwar immer von einer Komponente aus der Menge „weniger aussichtsreicher“ Komponenten  $L^-$  hin zu einer „aussichtsreichen“ Komponente aus  $L^+$ . Immer dann, wenn die Gefahr besteht, in einen Zyklus zu geraten, wird die Schrittweite (also die Menge des zu verschiebenden Budgets) gemäß der Vorschrift  $v = \frac{\hat{B}}{\mu}$  angepasst, wobei  $\mu$  bei jeder Anpassung um 1 erhöht wird. Durch diese Verringerung der Schrittweite  $v$  nähert man sich numerisch immer weiter einem KKT-Punkt an und dadurch, dass immer das gesamte Budget auf die Komponenten verteilt ist, bleibt das Verfahren (primal) zulässig. Zudem gilt für jede Iteration, dass sich entweder der Zielfunktionswert verbessert oder aber die Schrittweite verringert wird. Als Abbruchbedingung kann man beispielsweise ein  $\varepsilon > 0$  vorgeben, bei dessen Unterschreitung durch die Schrittweite, also für  $v < \varepsilon$ , das Verfahren abbricht. Ebenso ist möglich, die Unterschreitung einer Zielspannweite der relevanten Grenznutzen als Abbruchkriterium zu betrachten, also mit  $R := \max_{i \in \{i | x_i > 0\}} \gamma_i - \min_{i \in \{i | x_i > 0\}} \gamma_i$  ist das Abbruchkriterium  $R < \varepsilon$ .<sup>5</sup> Auch Kombinationen von Abbruchkriterien sind möglich, wie z. B.  $\max \{v, R\} < \varepsilon$ .

Die Iterationen laufen nun folgendermaßen ab: In Zeile 2 von Alg. 5.1 wird eine Iterationsmenge  $C'$  initialisiert und die Menge  $L^-$  der weniger aussichtsreichen Komponenten als die Menge der Komponenten mit minimalem Grenznutzen in der Iterationsmenge festgelegt. In Zeile 3 und 4 werden iterativ diejenigen Komponenten aus der Iterationsmenge  $C'$  entfernt, für die in der aktuellen Lösung  $x_i = 0$  gilt, und im Anschluss wird  $L^-$  aktualisiert, da sich diese Menge über die Menge  $C'$  berechnet.

In Zeile 5 wird die Schrittweite, also das Budget, das verschoben wird, unter bestimmten Bedingungen verringert. In Zeile 6 wird Geld in Höhe der Schrittweite  $v$  einer Komponente aus der Menge der weniger aussichtsreichen Komponenten „zur Seite gelegt“, in Zeile 7 die Menge der aussichtsreichen Komponenten  $L^+$  aktualisiert und das zur Seite gelegte Geld in Zeile 8 wieder investiert. Aufgrund des abnehmenden Grenznutzens der finanziellen Mittel wird es im Verlauf des Algorithmus häufiger dazu kommen, dass das Geld in dieselbe Komponente reinvestiert wird, wenn sie aus  $L^-$  nach  $L^+$  gewechselt ist. Hier sorgt Zeile 5 also dafür, dass der Algorithmus nicht dauerhaft in einem Zyklus verbleibt.

Ein exemplarischer Vergleich der Implementierung der Grenznutzenmethode Alg. 5.1 mit

---

<sup>5</sup>Ein Vorteil der Einbeziehung dieser Spannweite wäre die Sicherstellung des hinreichenden „Einpendelns“ auf den Lagrangemultiplikator  $\lambda$  (vgl. Abschnitt 5.3.1).

einem Standardsolver für globale Optimierung für das einperiodige Erwartungswertmodell hinsichtlich der Anzahl an benötigten Iterationen findet sich in Abschnitt 7.2.2.

Die weitere Prüfung, inwieweit das hier zunächst für den einperiodigen Spezialfall entwickelte Verfahren auch ein globales Optimum liefern kann, geschieht im Folgenden im Rahmen der Analyse des mehrperiodigen Erwartungswertmodells.

## 5.4 Analyse des mehrperiodigen Erwartungswertmodells

Wir wollen Hinweise dafür finden, in welchen Fällen eine effiziente Lösung des Problems aufgrund der Eigenschaften der Zielfunktion möglich ist. Dazu können wir zwar nicht auf alle Methoden der konvexen (bzw. konkaven) Optimierung (s. bspw. Boyd u. a. (2009)) zurückgreifen, aber für die Oberklasse der *quasikonkaven* Optimierungsprobleme (s. bspw. Avriel u. a. (2010)) kann unter bestimmten Bedingungen ebenfalls eine effiziente und global optimale Lösung gefunden werden. Daher wollen wir in Abschnitt 5.4.1 für einen Spezialfall die globale Optimalität herausarbeiten und in Abschnitt 5.4.2 Überlegungen zur Übertragbarkeit auf allgemeine Strukturbewertungsfunktionen sowie zum Lösungsverfahren anstellen.



### 5.4.1 Globale Optimalität quasikonkaver Zielfunktionen für einen Spezialfall

Wir wollen zeigen, dass ein KKT-Punkt unter bestimmten Voraussetzungen das globale Optimum für den mehrperiodigen Fall darstellt. Betrachten wir dafür das mehrperiodige Erwartungswertmodell in folgender Form:

$$\text{Max.}_{\mathbf{x}} \min_{t=1,\dots,T} \phi_t(\mathbf{x}) \quad (5.29)$$

u. d. N.

$$\sum_{t=1}^T \sum_{i=1}^n x_{it} = \hat{B} \quad (5.30)$$

$$x_{it} \geq 0 \quad (i = 1, \dots, n; t = 1, \dots, T)$$

$$p_{i0} = \hat{p}_i \quad (i = 1, \dots, n) \quad (5.31)$$

mit  $\phi_t(\mathbf{x}) = \varphi^{\text{Ser}}(\mathbf{g}(\mathbf{x}_t, \mathbf{p}_{t-1}, \widehat{\Delta \mathbf{w}}_t))$ . Als Spezialfall betrachten wir dabei die Strukturbewertungsfunktion für ein Seriensystem

$$\varphi^{\text{Ser}}(\mathbf{p}) = \prod_{i=1,\dots,n} p_i$$

und nehmen an, dass  $\widehat{\Delta w}_{it} < p_{i,t-1}$ , woraus die Konkavität der Zustandsübergangsfunktionen  $g_i$  resultiert.

Für diesen Spezialfall wollen wir folgendermaßen vorgehen:

1. Wir zeigen die Quasikonkavität der komponierten Systemleistungsfunktion  $\phi_t := \varphi \circ \mathbf{g}$  in  $\mathbf{x}$  für festes  $t = 1, \dots, T$ .
2. Wir zeigen die Quasikonkavität der Zielfunktion  $f(\mathbf{x}) := \min_{t=1,\dots,T} \phi_t(\mathbf{x})$ .
3. Wir schlussfolgern die globale Optimalität für einen KKT-Punkt des mehrperiodigen Erwartungswertmodells.

## Quasikonkavität der Systemleistungsfunktion

Zunächst definieren wir, was quasikonkav bedeutet.

**Definition 4** (in Anlehnung an Avriel u. a. (2010, S. 57)). Sei  $C \subset \mathbb{R}^n$  eine konvexe Menge. Dann ist eine Funktion  $h : C \rightarrow \mathbb{R}$  quasikonkav genau dann, wenn aus  $\mathbf{x}', \mathbf{x}'' \in C$  und  $0 \leq \theta \leq 1$  folgt, dass

$$\min \{h(\mathbf{x}'), h(\mathbf{x}'')\} \leq h((1 - \theta) \cdot \mathbf{x}' + \theta \cdot \mathbf{x}'').$$

Bei quasikonkaven Funktionen darf der Funktionswert aller Punkte auf einer Verbindungsline von zwei Punkten also nicht kleiner sein als der kleinere der beiden Funktionswerte dieser Punkte selbst.

Als nächstes wollen wir zeigen, dass die Systemleistungsfunktion als Komposition der ausgewählten Strukturbewertungsfunktion mit einer Zustandsübergangsfunktion quasikonkav ist. Wir wenden dafür den folgenden Satz an:

**Satz 5** (in Anlehnung an Avriel u. a. (2010, S. 162)). Sei  $C \subset \mathbb{R}^n$  eine konvexe Menge und  $G_i : C \rightarrow \mathbb{R}_{\geq 0}$  seien nichtnegative konkave Funktionen. Dann ist die zusammengesetzte Funktion

$$\prod_{i=1}^n G_i(\mathbf{x})$$

quasikonkav auf  $C$ .

Daraus folgern wir

**Korollar 6.** Die Systemleistungsfunktion  $\phi_t := \varphi^{\text{Ser}} \circ \mathbf{g}$  ist für festes  $t = 1, \dots, T$  quasikonkav.

*Beweis.* Seien die Funktionen  $G_i$  definiert über die Zustandsübergangsfunktionen, also  $G_i(\mathbf{x}) := g_i(x_{it}, p_{i,t-1}, \widehat{\Delta w_{it}})$ . Dann ist für festes  $t = 1, \dots, T$

$$\phi_t(\mathbf{x}) = \varphi^{\text{Ser}}(\mathbf{G}(\mathbf{x})) = \prod_{i=1}^n g_i(x_{it}, p_{i,t-1}, \widehat{\Delta w_{it}}).$$

Da die Zustandsübergangsfunktionen sowohl nichtnegativ und konkav als auch auf einer konvexen Menge definiert sind, ist die Systemleistungsfunktion nach Satz 5 quasikonkav.  $\square$

## Quasikonkavität der Zielfunktion

Nun möchten wir für den Spezialfall die Quasikonkavität der Zielfunktion zeigen, die im Erwartungswertmodell maximiert werden soll, nämlich die minimale Perioden-Systemleistung. Nach Volle (1997, S. 365) ist das Maximum zweier quasikonvexer Funktionen wieder quasikonvex. Daraus folgern wir

**Korollar 7.** *Das Minimum zweier quasikonkaver Funktionen ist quasikonkav.*

*Beweis.* Nach Boyd u. a. (2009, S. 95) ist eine Funktion  $f$  genau dann quasikonkav, wenn  $-f$  quasikonvex ist. Betrachte das Minimum zweier quasikonkaver Funktionen  $f_1$  und  $f_2$ , die auf einer konvexen Menge  $C \subset \mathbb{R}^n$  definiert sind. Es gilt

$$\min \{f_1(\mathbf{x}), f_2(\mathbf{x})\} = - \underbrace{\max \{-f_1(\mathbf{x}), -f_2(\mathbf{x})\}}_{=:k(\mathbf{x})}$$

Da nach Volle (1997, S. 365)  $k(\mathbf{x})$  quasikonvex ist, ist  $-k(\mathbf{x})$  quasikonkav.  $\square$

Für den interessierten Leser sei ein Beweis für strikte Quasikonkavität in Anhang A.4 gegeben.

Mit diesem Ergebnis können wir folgendes Korollar anschließen:

**Korollar 8.** *Die Zielfunktion des Erwartungswertmodells*

$$f(\mathbf{x}) = \min_{t=1,\dots,T} \phi_t(\mathbf{x}) = \min_{t=1,\dots,T} \left( \varphi^{Ser} \left( \left( g_i(x_{it}, p_{i,t-1}, \widehat{\Delta w}_{it}) \right)_{i=1,\dots,n} \right) \right)_{t=1,\dots,T}$$

*ist quasikonkav.*

*Beweis.* Die Entscheidungsvariablen  $\mathbf{x}$  sind auf der Menge

$$\mathcal{X} = \left\{ x_{it} \mid x_{it} \geq 0, (i = 1, \dots, n; t = 1, \dots, T), \sum_{t=1}^T \sum_{i=1}^n x_{it} = \widehat{B} \right\}$$

definiert, die konvex ist. Da sich das Prinzip

$$\min \{ \phi_1(\mathbf{x}), \min \{ \phi_2(\mathbf{x}), \phi_3(\mathbf{x}) \} \} = \min \{ \phi_1(\mathbf{x}), \phi_2(\mathbf{x}), \phi_3(\mathbf{x}) \}$$

induktiv auf  $T \geq 3$  Perioden anwenden lässt, ist die Zielfunktion  $f(\mathbf{x}) = \min_{t=1,\dots,T} \phi_t(\mathbf{x})$  mit Hilfe von Korollar 7 quasikonkav.  $\square$

### Globale Optimalität eines KKT-Punktes

Um die globale Optimalität für den Spezialfall abzuleiten, nutzen wir erneut ein Ergebnis aus der Literatur. Die nachfolgende Formulierung über quasikonkave Funktionen  $\mathbf{q}(\mathbf{x})$  ist gleichbedeutend damit, dass die *Constraint Qualification* für das Problem erfüllt ist.

**Satz 9** (in Anlehnung an Arrow und Enthoven (1961)). *Sei  $f(\mathbf{x})$  eine differenzierbare quasikonkave Funktion in einem  $n \cdot T$ -dimensionalen Vektor  $\mathbf{x}$ . Sei weiterhin  $\mathbf{q}(\mathbf{x})$  eine  $m$ -dimensionale differenzierbare quasikonkave Vektorfunktion.  $f$  und  $\mathbf{q}$  seien für  $\mathbf{x} \geq \mathbf{0}$  definiert. Betrachte einen KKT-Punkt  $(\bar{\mathbf{x}}, \bar{\lambda})$  des mehrperiodigen Erwartungswertmodells Gl. (5.29) – (5.31), für den mindestens eine der folgenden Bedingungen erfüllt ist:*

- 1) *Es gibt eine Kombination  $(i', t')$ , für die  $\frac{\partial}{\partial \bar{x}_{i't'}} f(\bar{\mathbf{x}}) > 0$  und  $\bar{x}_{i't'} > 0$ .*
- 2)  *$\nabla f(\bar{\mathbf{x}}) \neq \mathbf{0}$ , und alle zweiten partiellen Ableitungen von  $f(\mathbf{x})$  an der Stelle  $\mathbf{x} = \bar{\mathbf{x}}$  existieren.*

*Dann ist  $\bar{\mathbf{x}}$  ein globales Maximum unter den Nebenbedingungen  $\mathbf{q}(\mathbf{x}) \geq \mathbf{0}, \mathbf{x} \geq \mathbf{0}$ .*

Aus dem Satz folgt:

**Korollar 10.** *Eine Lösung  $\bar{\mathbf{x}}$  des mehrperiodigen Erwartungswertmodells Gl. (5.29) – (5.31), die die KKT-Bedingungen erfüllt, ist global optimal, wenn es mindestens eine Kombination  $(i', t')$  gibt, für die  $\frac{\partial}{\partial \bar{x}_{i't'}} f(\bar{\mathbf{x}}) > 0$  und  $\bar{x}_{i't'} > 0$  oder wenn  $\nabla f(\bar{\mathbf{x}}) \neq \mathbf{0}$  und alle zweiten partiellen Ableitungen von  $f(\mathbf{x})$  an der Stelle  $\mathbf{x} = \bar{\mathbf{x}}$  existieren.*

*Beweis.* Die Nebenbedingung Gl. (5.30) aus dem Erwartungswertmodell formulieren wir zunächst um.  $\sum_{t=1}^T \sum_{i=1}^n x_{it} = \hat{B}$  wird aufgeteilt in die beiden gleichwertigen Nebenbedingungen

$$q_1(\mathbf{x}) := \hat{B} - \sum_{t=1}^T \sum_{i=1}^n x_{it} \geq 0$$

$$q_2(\mathbf{x}) := \sum_{t=1}^T \sum_{i=1}^n x_{it} - \hat{B} \geq 0$$

Die Funktionen  $\mathbf{q}(\mathbf{x}) = (q_m(\mathbf{x}))_{m=1,2}$  sind aufgrund ihrer Linearität (quasi-)konkav. Zur Gewährleistung der Differenzierbarkeit der Zielfunktion  $f(\mathbf{x}) = \min_{t=1,\dots,T} \phi_t(\mathbf{x})$  kann für eine Implementierung auf einem Digitalrechner eine bekannte Näherungsformel für das Minimum genutzt werden (vgl. Drud (2021b)):

$$\begin{aligned} \min \{ \phi_1(\mathbf{x}), \phi_2(\mathbf{x}) \} &= \frac{\phi_1(\mathbf{x}) + \phi_2(\mathbf{x}) - |\phi_1(\mathbf{x}) - \phi_2(\mathbf{x})|}{2} \\ &\approx \frac{\phi_1(\mathbf{x}) + \phi_2(\mathbf{x}) - \sqrt{(\phi_1(\mathbf{x}) - \phi_2(\mathbf{x}))^2 + \delta^2}}{2} \end{aligned}$$

mit  $\delta > 0$  beliebig klein. Durch die Bedingungen 1) und 2) des Satzes 9, die auch im Korollar vorausgesetzt werden, ist jeder KKT-Punkt des Spezialfalls des mehrperiodigen Erwartungswertmodells ein globales Optimum.  $\square$

Die Voraussetzungen des Korollars 10 müssen für die konkrete Instanziierung des Erwartungswertmodells jeweils geprüft werden. Für die Performance-Analyse der Seriensystem-Instanzen in Kapitel 7 sind diese Voraussetzungen erfüllt. Eine genauere Analyse der KKT-Bedingungen für den mehrperiodigen Fall findet sich in Anhang A.3.4.

### 5.4.2 Lösungsmethodik für allgemeine Strukturbewertungsfunktionen

Wir möchten nun zunächst die Übertragbarkeit des Spezialfalls auf allgemeinere Strukturbewertungsfunktionen untersuchen, bevor wir die Lösungsmethodik besprechen.

#### Übertragbarkeit auf allgemeine Strukturbewertungsfunktionen

In dem Fall, dass nicht nur monotone, sondern auch quasikonkave Strukturbewertungsfunktionen vorliegen, kann die Übertragbarkeit gezeigt werden. Zu diesem Zweck möchten wir folgenden Satz anwenden, der sich auf *quasikonvexe* Funktionen bezieht.

**Satz 11** (in Anlehnung an Agrawal und Boyd (2020)). *Sei  $\psi : \mathbb{R}^k \supset C_1 \rightarrow \mathbb{R}$  eine quasikonvexe Funktion und  $\{P_1, P_2, P_3\}$  eine Partition von  $\{1, \dots, n\}$ , sodass  $\psi$  nichtfallend in den Argumenten ist, die durch  $P_1$  indiziert sind, und nichtwachsend in den Argumenten, die durch  $P_2$  indiziert sind. Habe weiterhin die Funktion  $\mathbf{g} : \mathbb{R}^n \supset C_2 \rightarrow \mathbb{R}^k$  die*

Eigenschaft, dass die  $g_i$  konvex für  $i \in P_1$ , konkav für  $i \in P_2$  und affin für  $i \in P_3$  sind. Dann ist die Komposition  $\psi \circ \mathbf{g}$  quasikonvex.

Aus diesem Satz können wir folgendes Korollar ableiten.

**Korollar 12.** Für eine quasikonkave Strukturbewertungsfunktion  $\varphi$  ist die Funktion  $\phi_t = \varphi \circ \mathbf{g} : \mathbb{R}_{\geq 0}^n \rightarrow \mathbb{R}, \mathbf{x} \mapsto \varphi \left( \left( g_i \left( x_{it}, p_{i,t-1}, \widehat{\Delta w}_{it} \right) \right)_{i=1,\dots,n} \right)$  quasikonkav für alle  $t = 1, \dots, T$ .

*Beweis.* Seien zunächst  $k = n$ ,  $C_1 := [0; 1]^n$  und  $C_2 := \mathbb{R}_{\geq 0}^n$ . Dann sind die Zustandsübergangsfunktionen  $g_i \left( x_{it}, p_{i,t-1}, \widehat{\Delta w}_{it} \right)$  für alle festen Zeitpunkte  $t$  und für alle Komponenten  $i$  konkav (s. Einleitung des Abschnitts 5.4.1), wenn  $p_{i,t-1} > \widehat{\Delta w}_{it}$ . Für die Partition  $\{P_1, P_2, P_3\}$  von  $\{1, \dots, n\}$  gelte  $P_2 = \{1, \dots, n\}$  und  $P_1 = P_3 = \emptyset$ . Weiterhin ist die Funktion  $\psi := -\varphi : [0; 1]^n \rightarrow \mathbb{R}, \mathbf{p} \mapsto -\varphi(\mathbf{p})$  quasikonvex, da die Strukturbewertungsfunktion  $\varphi$  gemäß Voraussetzung quasikonkav ist. Zudem ist  $\psi$  nichtwachsend in den Argumenten, die durch  $P_2$  indiziert sind, da  $\varphi$  aufgrund der Monotonie der Strukturbewertungsfunktion nichtfallend ist.

Die Anwendung des Satzes 11 führt dazu, dass  $(-\varphi) \circ \mathbf{g}$  quasikonvex ist, also ist  $\phi_t = \varphi \circ \mathbf{g}$  quasikonkav für alle Zeitpunkte  $t = 1, \dots, T$ .  $\square$

Es konnte für die Voraussetzung, dass eine Strukturbewertungsfunktion zwar monoton wachsend, aber nicht quasikonkav ist, bisher kein Beweis für eine generelle Übertragbarkeit erbracht werden. Obwohl für den eindimensionalen Fall z. B. in Avriel u. a. (2010, S. 57 f.) gezeigt wurde, dass die monotone Transformation einer (quasi-)konkaven Funktion quasikonkav ist, konnten für das Vorliegen einer mehrdimensionalen Strukturbewertungsfunktion keine Resultate gefunden werden. Wäre dies der Fall, so könnten die Überlegungen des Abschnitts 5.4.1 zur Zielfunktion bzw. zur Optimalität im KKT-Punkt analog auf monotone Strukturbewertungsfunktionen übertragen werden.

## Lösungsmethodik

Wir greifen wir zur Lösung des Problems auf einen bewährten Optimierungssolver zurück. Der Solver CONOPT3 (vgl. Drud (2020)) ist ein GRG- („generalized reduced gradient“-)basierter Solver für große nichtlineare Optimierungsprobleme (NLP), der KKT-Punkte bestimmt und dadurch lokale Optima findet. Da ein lokales Optimum zumindest für

den behandelten Spezialfall bewiesenermaßen eine global optimale Lösung ist, kann also mit Hilfe dieses Solvers wie in Abschnitt 4.4 beschrieben eine rollierende Planung des mehrperiodigen Erwartungswertmodells durchgeführt werden.

Eine ausführliche Beschreibung des Algorithmus, der CONOPT3 zugrundeliegt, sowie Performance-Vergleiche mit anderen nichtlinearen Solvern findet man z. B. bei Andrei (2017b, S. 331 ff.). Eine beispielhafte Implementierung in der Modellierungssprache GAMS (General Algebraic Modeling System) für eine ausgewählte Instanz findet sich im Anhang B.2.1 auf Seite 168 f.

Wir müssen allerdings feststellen, dass eine monotone Strukturbewertungsfunktion i. A. nicht quasikonkav ist.<sup>6</sup> Es muss daher davon ausgegangen werden, dass selbst für den einperiodigen Fall mehrere KKT-Punkte existieren können (außer für Spezialfälle wie das Seriensystem). Im einperiodigen Fall liefert die Grenznutzenmethode per Konstruktion immer einen KKT-Punkt.

Bei numerischen Untersuchungen, die im Rahmen des Abschnitts 7.2.1 für alle in dieser Arbeit betrachteten Instanzen durchgeführt wurden, kam im Vergleich mit dem exakten Solver LindoGlobal heraus, dass der CONOPT3-Solver, der u. a. auf der Auswertung der KKT-Bedingungen beruht, auch für den mehrperiodigen Fall stets eine optimale Lösung geliefert hat. Es wird daher vermutet, dass die Anwendung des CONOPT3-Solvers für die in dieser Arbeit untersuchten Systeme, Zustandsübergangsfunktionen und Verschleißprozesse innerhalb kurzer Rechenzeiten gute Ergebnisse liefert, für die allerdings nach aktuellem Stand keine Optimalität garantiert werden kann. Im Ergebnis ist allerdings die rollierende Planung, in deren Anwendungskontext der Solver eingesetzt wird, nur eine Heuristik, für die aufgrund stochastischer Verschleißprozesse ohnehin keine Optimalität gewährleistet werden kann.

---

<sup>6</sup>Ein Gegenbeispiel ist bspw. ein zweikomponentiges Parallelsystem.

# 6 Stochastisches dynamisches Optimierungsmodell für die flexible Planung der Budgetallokation

Wir möchten nun den in Abschnitt 2.5.2 angesprochenen alternativen Ansatz der flexiblen Planung näher beleuchten. Grundlage dafür ist die Formulierung des Planungsproblems als stochastisches dynamisches Optimierungsmodell. Damit beschäftigen wir uns in Abschnitt 6.1. In Abschnitt 6.2 werden wir verschiedene Lösungsansätze vorstellen, die uns zu dem in dieser Arbeit genutzten Ansatz des Deep Reinforcement Learning führen. In Abschnitt 6.3 stellen wir zuletzt dar, wie die konkrete Umsetzung der flexiblen Planung mit Hilfe neuronaler Netze aussehen wird.

## 6.1 Formulierung und Interpretation des stochastischen dynamischen Optimierungsproblems

Wir möchten unser Modell nun ähnlich zum Erwartungswertmodell aus Abschnitt 4.4 formulieren und gehen dabei auf die Besonderheiten ein, die sich durch die Verwendung von Zufallsvariablen ergeben. Die Tilde deutet dabei auf den stochastischen Charakter der Größen hin,  $\widetilde{\Delta w}$  z. B. ist also eine Zufallsvariable im folgenden stochastischen Modell.



$$\text{Max.}_{\mathbf{x}} \min_{t=1, \dots, T} \tilde{\phi}_t \quad (6.1)$$

u. d. N.

$$\tilde{\phi}_t = \varphi((\tilde{p}_{it})_{i=1, \dots, n}) \quad (t = 1, \dots, T) \quad (6.2)$$

$$\tilde{p}_{it} = g_i(x_{it}, \tilde{p}_{i,t-1}, \widetilde{\Delta w_{it}}) \quad (i = 1, \dots, n; t = 1, \dots, T) \quad (6.3)$$

$$B_t = B_{t-1} - \sum_{i=1}^n x_{it} \quad (t = 1, \dots, T)$$

$$B_t \geq 0 \quad (t = 1, \dots, T)$$

$$x_{it} \geq 0 \quad (i = 1, \dots, n; t = 1, \dots, T)$$

$$B_0 = \hat{B}$$

$$p_{i0} = \hat{p}_i \quad (i = 1, \dots, n)$$

Aufgrund des Vorhandenseins von Zufallsvariablen muss dieses Modell interpretiert werden. Dem Verschleiß, der wie für das Erwartungswertmodell als gammaverteilt angenommen wird (vgl. Abschnitte 2.4 und 4.2), kommt eine besondere Bedeutung zu. Weil in Nebenbedingung (6.3) die Größe  $\widetilde{\Delta w_{it}}$  eine Zufallsvariable ist, also erst durch Realisation einen konkreten Wert annimmt, sind die Komponentenzustände  $\tilde{p}_{it}$  am Ende der Periode  $t$  ebenfalls Zufallsvariablen. Dieser Umstand wirkt sich auf die Systemleistung  $\tilde{\phi}_t$  am Ende der Periode  $t$  aus (s. Gl. (6.2)) und das wiederum auf die Zielfunktion in Gl. (6.1).

Die Zustandsübergangsfunktion  $g_i$  einer Komponente  $i$  sei nun bezüglich der konkreten Realisation  $\widehat{\Delta w_{it}}$  von  $\widetilde{\Delta w_{it}}$  bzw.  $p_{i,t-1}$  von  $\tilde{p}_{i,t-1}$  ebenso definiert wie für das Erwartungswertmodell in Abschnitt 4.3.1, also

$$g_i : \mathbb{R}_{\geq 0} \times [0; 1] \times \mathbb{R}_{\geq 0} \longrightarrow [0; 1], (x, p, \Delta w) \mapsto g_i(x, p, \Delta w).$$

$g_i$  lasse sich wieder zerlegen, also mit der Investitionsfunktion  $g_i^+$  gelte

$$g_i(x_{it}, p_{i,t-1}, \widehat{\Delta w_{it}}) = \max \{g_i^+(x_{it}, p_{i,t-1}) - \widehat{\Delta w_{it}}; 0\}$$

und für  $g_i^+$  gelte erneut der abnehmende Grenznutzen des finanziellen Mitteleinsatzes

(vgl. auch Abschnitte 2.2.2 bzw. 4.3.1) sowie

$$\lim_{x_{it} \rightarrow \infty} g_i^+(x_{it}, p_{i,t-1}) = 1.$$

Dadurch, dass die Gamma-Verteilung nur über der Menge der nichtnegativen reellen Zahlen definiert ist (es gilt für die Realisationen  $\widehat{\Delta w}_{it} \geq 0$ ), ist der Wertebereich der Zustandsübergangsfunktion somit weiterhin auf den Bereich  $[0; 1]$  beschränkt.

Ebenso kann die Strukturbewertungsfunktion  $\varphi$  bezüglich der Realisationen  $p_{it}$  von  $\tilde{p}_{it}$  definiert werden (vgl. Abschnitt 4.3.2):

$$\varphi : [0; 1]^n \longrightarrow [0; 1], \mathbf{p}_t \mapsto \varphi(\mathbf{p}_t).$$

Somit ist bei der Zielfunktion (vgl. Abschnitt 4.3.3) das Minimum über die Perioden-Systemleistungen ebenfalls bezüglich der Realisationen definiert. Da das stochastische Programm somit über Realisationen entwickelt wird, wollen wir zusammenfassen, welche Realisationen für Zielfunktion, Nebenbedingungen und Lösung zugrundegelegt werden. Bei der Zielfunktion sind dies die Realisationen, die den erwarteten Zielfunktionswert liefern, bei den Nebenbedingungen sind dies alle Realisationen und für die Lösung spezifiziert jede Realisation  $\widehat{\Delta w}$ , die bei Anwendung einer optimalen flexiblen Politik  $\pi$  eintreten kann, ein Investitionsprogramm  $\mathbf{x}$ . Auf den Begriff der Politik werden wir gleich noch einmal genauer eingehen.

Da es sich bei den Argumenten der Zielfunktion um Zufallsvariablen handelt, muss sich die Optimierung auf Kennzahlen der Verteilung der Zielfunktionswerte wie Lage- oder Dispersionsmaß beziehen. Wir wählen die häufigste Form der Operationalisierung mit Bildung des Erwartungswerts über die Gesamtfunktion

$$\text{Max.}_{\mathbf{x}} f(\mathbf{x}) := \mathbb{E} \left( \min_{t=1, \dots, T} \tilde{\phi}_t \right). \quad (6.4)$$

Prinzipiell könnten auch weitere Kennzahlen wie die Varianz dieser Zufallsvariablen in die Zielfunktion einbezogen werden.

Nun müssen wir eine Interpretation für die Lösung finden, die für jede Realisation  $(\widehat{\Delta w}_{it})_{i,t}$  der Verschleißprozesse  $\{\widehat{\Delta w}_{it} \mid t = 1, \dots, T\}$  ein Investitionsprogramm  $(\mathbf{x}_t)_t$  liefert mit  $\mathbf{x}_t$  als Funktionen  $\pi_t(\mathbf{p}_{t-1}, B_{t-1})$  des Zustands  $\mathbf{p}_{t-1}$  und des Restbudgets  $B_{t-1}$

am Ende der Periode  $t - 1$  so, dass der resultierende Erwartungswert  $\mathbb{E} \left( \min_{t=1, \dots, T} \tilde{\phi}_t \right)$  maximiert wird.

Wie bei der rollierenden Planung behandeln wir ein dynamisches Modell, bei dem sich Festlegungen und Realisationen im Zeitverlauf abwechseln. In der flexiblen Planung wird allerdings grundsätzlich nicht zu Beginn jeder Periode neu geplant, sondern es werden im Voraus Eventualpläne, also Handlungsempfehlungen in Abhängigkeit des jeweils beobachteten Zustands gesucht. Die Funktion der Handlungsempfehlungen in Abhängigkeit eines Zustands nennt man in der stochastischen dynamischen Optimierung *Strategie* oder *Politik*, die häufig mit  $\pi$  bezeichnet wird und im vorliegenden Fall eine Rückkopplungssteuerung darstellt.<sup>1</sup> Im besten Fall findet man eine optimale Politik  $\pi^* : \mathcal{S} \rightarrow \mathcal{X}$ , die für jeden Zustand  $\mathbf{s} \in \mathcal{S}$  eine optimale Aktion  $\mathbf{x}^* \in \mathcal{X}(\mathbf{s})$  liefert, mit  $\mathcal{S} = \times_{t=1}^T \mathcal{S}_t$ ,

$$\mathcal{S}_t \subseteq \{(\mathbf{p}_{t-1}, B_{t-1}) \mid \mathbf{p}_{t-1} \in [0; 1]^n, 0 \leq B_{t-1} \leq \hat{B}\}$$

und  $\mathcal{X} = \cup_{\mathbf{s} \in \mathcal{S}} \mathcal{X}(\mathbf{s})$ ,  $\mathbf{s} = (\mathbf{s}_1, \dots, \mathbf{s}_T) \in \mathcal{S}$ ,  $\mathcal{X}(\mathbf{s}) = \times_{t=1}^T \mathcal{X}_t(\mathbf{s}_t)$  mit

$$\mathcal{X}_t(\mathbf{s}_t) = \left\{ \mathbf{x} \mid x_{it} \geq 0, \sum_{i=1}^n x_{it} \leq B_{t-1} \right\}.$$

Daher wird das Ziel der Optimierung anstelle von Gl. (6.4) präziser mit

$$\text{Max.}_{\pi} \mathbb{E} \left( \min_{t=1, \dots, T} \tilde{\phi}_t \right)$$

beschrieben, um so auszudrücken, dass man eine optimale Politik anstelle eines Entscheidungsvektors sucht. Da das Reinforcement Learning<sup>2</sup> diverse Ansätze für die Ermittlung solcher Politiken bietet, möchten wir uns im Folgenden dessen Methoden bedienen.

<sup>1</sup>Politiken können bei kleinen Zustands- und Aktionenräumen in Form von Entscheidungsbäumen dargestellt werden, vgl. Abschnitt 2.5.2.

<sup>2</sup>Die deutsche Übersetzung „bestärkendes Lernen“ wird selten verwendet, weswegen wir bei der Verwendung des englischen Begriffs bleiben.

## 6.2 Ausgewählte Methoden des Reinforcement Learning

Grundlage für die Methoden des Reinforcement Learning ist die Bellmansche Funktionalgleichung (s. Bellman (1966)), die ursprünglich für die deterministische Variante der dynamischen Optimierung entwickelt wurde. Bei dieser Funktionalgleichung setzt sich die Zielfunktion auf jeder (Zeit-)Stufe  $t$  zusammen aus einer Belohnungsfunktion (engl. „reward“, vgl. Richter (2019, S. 257))

$$r : \mathcal{S}_t \times \mathcal{X}_t \times \mathcal{S}_{t+1} \longrightarrow \mathbb{R}, (\mathbf{s}_t, \mathbf{x}_t, \tilde{\mathbf{s}}_{t+1}) \mapsto r(\mathbf{s}_t, \mathbf{x}_t, \tilde{\mathbf{s}}_{t+1})$$

sowie einer Wertfunktion (engl. „value function“)  $V_{t+1}(\mathbf{s}_{t+1})$ . Dabei ist  $\mathcal{S}_t$  der Definitionsbereich der Zustandsvariablen und  $\mathcal{X}_t$  der Definitionsbereich der Entscheidungsvariablen zum Zeitpunkt  $t$ . Die Belohnungsfunktion hängt vom Zustand  $\mathbf{s}_t$ , einer Aktion  $\mathbf{x}_t$  und ggf. vom unsicheren Folgezustand  $\tilde{\mathbf{s}}_{t+1}$  ab, d. h., die Belohnung wird erst nach Realisation von  $\tilde{\mathbf{s}}_{t+1}$  sichtbar. Die Wertfunktion  $V_{t+1}(\mathbf{s}_{t+1})$  bemisst den Gesamtwert einer optimalen Politik ab Stufe  $t + 1$  bei gegebenem Anfangszustand  $\mathbf{s}_{t+1}$ . Der Wert des aktuellen Zustands  $V_t(\mathbf{s}_t)$  ist somit der optimale Zielfunktionswert für das Teilproblem ab Stufe  $t$  für gegebenen Zustand  $\mathbf{s}_t$ .

Die bekannte Rekursionsgleichung lautet für stochastische dynamische Programme (vgl. z. B. Powell (2011, S. 60))

$$V_t(\mathbf{s}_t) = \max_{\mathbf{x}_t \in \mathcal{X}(\mathbf{s}_t)} \left\{ \mathbb{E}(r(\mathbf{s}_t, \mathbf{x}_t, \tilde{\mathbf{s}}_{t+1})) + \gamma \cdot \underbrace{\int_{\mathbf{s}' \in \mathcal{S}_{t+1}} \mathbb{P}(\tilde{\mathbf{s}}_{t+1} = \mathbf{s}' \mid \mathbf{s}_t, \mathbf{x}_t) \cdot V_{t+1}(\mathbf{s}') d\mathbf{s}'}_{=: \mathbb{E}(V_{t+1}(\tilde{\mathbf{s}}_{t+1}) \mid \mathbf{s}_t, \mathbf{x}_t)} \right\} \quad (6.5)$$

mit einer zustandsabhängigen Menge der Aktionen  $\mathcal{X}(\mathbf{s}_t)$  und einem Diskontierungsfaktor  $\gamma \in [0; 1]$ , der angibt, ob eine rein myopische Strategie verfolgt wird ( $\gamma = 0$ ), ob alle Folgestände gleich gewichtet werden ( $\gamma = 1$ ) oder eine Gewichtung dazwischen erfolgt. In dieser Arbeit setzen wir  $\gamma := 1$ . Voraussetzung für diese Formulierung ist, dass ein zeitdiskreter Markovscher Entscheidungsprozess vorliegt.

Da bei uns die Verschleiße  $\widetilde{\Delta w_{it}}$  für festes  $i$  stochastisch unabhängig sind und sich Zustand  $p_{it}$  eindeutig über die Zustandsübergangsfunktion  $g_i$  eindeutig aus Zustand  $p_{i,t-1}$ ,

Investition  $x_{it}$  und realisiertem Verschleiß  $\Delta w_{it}$  ergibt, liegt bei uns ein Markovscher Entscheidungsprozess vor. Damit gilt insgesamt für den optimalen Zielfunktionswert eines stochastischen dynamischen Programms unter einem (zeitdiskreten) Markovschen Entscheidungsprozess<sup>3</sup>  $\{(\mathbf{s}_t, \mathbf{x}_t) \mid t = 1, \dots, T\}$  mit Politik  $\pi$  (vgl. Richter (2019, S. 260))

$$V_1^\pi(\mathbf{s}) = \mathbb{E} \left( \sum_{t=1}^T r(\tilde{\mathbf{s}}_t, \pi(\tilde{\mathbf{s}}_t), \tilde{\mathbf{s}}_{t+1}) \mid \hat{\mathbf{s}}_1 = \mathbf{s} \right). \quad (6.6)$$

Um besser nachzuvollziehen, dass sich auch unser Modell eignet, um Methoden des Reinforcement Learning (s. bspw. Sutton und Barto (2018), die dazu einen aktuellen Überblick bieten) anwenden zu können, formulieren wir unsere Zielfunktion im späteren Abschnitt 6.3.1 im Sinne der Darstellung (6.6) um.

### 6.2.1 Politikiteration, Wertiteration und Actor-Critic-Verfahren

Wir wenden uns nun zunächst den Standard-Methoden zur Lösung Markovscher Entscheidungsprobleme – Politik- und Wertiteration – zu, von denen wir dann das in der vorliegenden Arbeit verwendete Actor-Critic-Verfahren abgrenzen, das im Zuge des Deep Reinforcement Learning eingesetzt werden wird.

Üblicherweise hat man es im Reinforcement Learning mit einer überschaubaren, endlichen Anzahl von möglichen Aktionen und Zuständen zu tun und nicht mit einem überabzählbaren Aktions- und Zustandsraum wie bei der vorliegenden Arbeit. Die Beschreibungen der Verfahren in diesem Abschnitt gehen daher von endlichen, diskreten Aktions- und Zustandsmengen aus, wobei wir zur Unterscheidung die diskreten Aktionen mit  $a$  anstelle von  $x$  bezeichnen.

In der Politikiteration (vgl. bspw. Sutton und Barto (2018, S. 80)) werden für alle Zustände  $s$  die Approximation der Wertfunktion  $\hat{V}(s)$  und die Politik  $\pi(s)$  mit beliebigen Werten initialisiert. Danach wechseln sich die Schritte „Politik-Evaluation“ und „Politik-Verbesserung“ solange ab, bis eine stabile Politik erreicht wird. Der Evaluationsschritt besteht darin, die Approximation der Wertfunktion  $\hat{V}(s)$  gemäß der aktuellen Politik  $\pi(s)$

---

<sup>3</sup>Im Folgenden wollen wir dies als Markovsches Entscheidungsproblem bezeichnen.

anzupassen, sodass

$$\hat{V}(s) \approx \sum_{s'} \sum_r \mathbb{P}(s', r \mid s, \pi(s)) \cdot (r + \gamma \hat{V}(s'))$$

mit Belohnungen  $r$ , wobei die Menge aller möglichen auftretenden Belohnungen als endlich angenommen wird.  $\mathbb{P}(s', r \mid s, \pi(s))$  stellt die bedingte Wahrscheinlichkeit dar, dass bei beobachtetem Zustand  $s$  und Politik  $\pi(s)$  der Folgezustand  $s'$  angenommen wird und gleichzeitig Belohnung  $r$  auftritt.

Der Verbesserungsschritt weist jedem Zustand  $s$  eine Politik  $\pi(s)$  gemäß

$$\pi(s) \in \arg \max_a \sum_{s'} \sum_r \mathbb{P}(s', r \mid s, a) \cdot (r + \gamma \hat{V}(s')),$$

also die bestmögliche Aktion zu. Falls sich die Politik im Verbesserungsschritt nicht ändert, ist sie stabil und das Verfahren beendet mit  $V(s) \approx \hat{V}(s)$ . Andernfalls wird mit der Politik  $\pi(s)$  erneut der Evaluationsschritt durchgeführt.

In der Wertiteration (vgl. bspw. Sutton und Barto (2018, S. 82 ff.)) hingegen wird für alle Zustände  $s$  nur die Approximation der Wertfunktion  $\hat{V}(s)$  mit beliebigen Werten initialisiert (außer auf der letzten Zeitstufe  $T$ , dort wird  $\hat{V}(s_T) = 0$  gesetzt). Es werden dann die Schritte der Politikiteration zusammengeführt, indem direkt das Maximum über alle zulässigen Aktionen gebildet wird, und solange iteriert wird, bis

$$V(s) \approx \max_a \sum_{s'} \sum_r \mathbb{P}(s', r \mid s, a) \cdot (r + \gamma V(s')).$$

Die Politik wird im Anschluss berechnet, mit

$$\pi(s) \in \arg \max_a \sum_{s'} \sum_r \mathbb{P}(s', r \mid s, a) \cdot (r + \gamma V(s')).$$

Manche Algorithmen führen Politik- oder Wertiteration nicht mit der Approximation der Wertfunktion  $\hat{V}(s)$ , sondern mit dem „Q-Wert“

$$Q(s, a) = \sum_{s' \in \mathcal{S}(s, a)} \mathbb{P}(s' \mid s, a) \cdot \left[ r(s, a, s') + \gamma \cdot \max_{a' \in \mathcal{A}(s')} Q(s', a') \right],$$

also einer Bewertungsfunktion für ein Zustands-Aktionspaar durch. Dabei ist  $\mathcal{A}(s)$  der

Aktionsraum in Abhängigkeit des Zustands  $s$ . Somit gilt die Beziehung

$$V(s) = Q(s, \pi^*(s)). \quad (6.7)$$

Politik- und Wertiteration können nur funktionieren, wenn die Anzahl der Aktionen und Zustände hinreichend klein ist (vgl. auch Abschnitt 2.5.2 zu den „Flächen der Dimensionalität“). Da die in dieser Arbeit betrachteten Aktions- und Zustandsräume überabzählbar viele Aktionen und Zustände beinhalten, wenden wir uns nun weiter entwickelten Verfahren zu.

Dies sind beispielsweise Actor-Critic-Verfahren, die zur Klasse der Politik-Gradienten-Methoden gehören (vgl. Sutton und Barto (2018, S. 321 ff.)). Politik-Gradienten-Methoden lernen eine parametrisierte Politik, die Aktionen prinzipiell ohne das Vorhandensein einer konkreten Wertfunktion auswählen kann. Eine Wertfunktion wird dann unter Umständen nur noch gebraucht, um die Politik-Parameter  $\theta$  zu lernen. Somit ist

$$\pi(a|s, \theta) = \mathbb{P}(a_t = a | s_t = s, \theta_t = \theta)$$

die Wahrscheinlichkeit, dass Aktion  $a$  zur Zeit  $t$  gewählt wird, wenn Zustand  $s$  und Parametervektor  $\theta$  zur Zeit  $t$  gegeben sind.<sup>4</sup> Es sei nun  $J(\theta)$  ein skalares Leistungsmaß bezüglich der Politik-Parameter. Dann versuchen Politik-Gradienten-Methoden dieses Leistungsmaß über Gradientenaufstieg

$$\theta_{t+1} = \theta_t + \alpha \cdot \nabla J(\theta_t)$$

mit Schrittweite  $\alpha$  zu maximieren, wobei  $\nabla J(\theta_t)$  eine Näherungsgröße für den tatsächlichen Gradienten darstellt. Actor-Critic-Verfahren lernen neben der parametrisierten Politik auch Näherungen für die Wertfunktion  $\hat{V}(s, \theta)$  oder den Q-Wert  $\hat{Q}(s, a, \theta)$ , wobei der Vektor  $\theta$  beispielsweise den Gewichten eines neuronalen Netzes entsprechen kann. Genau diesen Ansatz verfolgt die Variante des Deep Reinforcement Learning, die im Rahmen dieser Arbeit implementiert wurde.

---

<sup>4</sup>Denn nach Sutton und Barto (2018, S. 58) ist eine Politik formal eine Abbildung von Zuständen auf Wahrscheinlichkeiten der Auswahl jeder möglichen Aktion. Bei deterministischen Politiken ist die Wahrscheinlichkeit gleich Eins und man erhält somit eine eindeutige Abbildung des Zustands und der Parameter auf die auszuführende Aktion.

Die Bezeichnung Actor bezieht sich somit auf die zu lernende Politik, die Bezeichnung Critic auf die zu lernende Wertfunktion oder den zu lernenden Q-Wert.

### 6.2.2 Überblick über wichtige Literatur zu Deep Reinforcement Learning

Bevor wir zur Beschreibung des Actor-Critic-Verfahrens aus dem Bereich des Deep Reinforcement Learning kommen, das im Rahmen der Umsetzung der flexiblen Planung in dieser Arbeit eingesetzt wurde, wollen wir einen kurzen Überblick über wichtige Literaturquellen geben.

Der Zusatz „Deep“ in Deep Reinforcement Learning bezieht sich darauf, dass tiefe (engl. „deep“) neuronale Netze mit mehr als zwei Schichten Anwendung finden. Grundlagen zu neuronalen Netzen als ein Kerngebiet der „Künstlichen Intelligenz“ finden sich z. B. in Russell und Norvig (2016).

Den ersten großen Erfolg mit Deep Reinforcement Learning verzeichneten Mnih u. a. (2013, 2015), die das Verfahren des Deep-Q-Learning anwendeten, um gegen Computer erfolgreich ATARI-Spiele zu spielen. Die Idee des Deep-Q-Learning besteht darin, den Q-Wert mit einem neuronalen Netzwerk zu approximieren. Dieser Ansatz basiert auf dem Q-Learning (s. Watkins und Dayan (1992)), integriert aber die erfolgreichen Konzepte des Experience Replay (s. Lin (1993) oder Richter (2019, S. 271)) und von Zielnetzwerken. Experience Replay bedeutet, dass die Zustandsübergänge, die während des Trainings der neuronalen Netze beobachtet und abgespeichert werden, für das Lernen zufällig wieder aus einem Speicher entnommen werden. Zielnetzwerke sind prinzipiell Hilfsnetzwerke, die das Training entkoppeln. Nach einer bestimmten Anzahl von Schritten werden die Gewichte dieser Hilfsnetzwerke mit denen des eigentlichen neuronalen Netzes gleichgesetzt (oder graduell übertragen), um ein stabiles Laufverhalten des Algorithmus zu gewährleisten.

Das Verfahren von Mnih u. a. (2015) konnte in seiner ursprünglichen Form nicht ohne Informationsverlust durch Diskretisierung auf das in dieser Arbeit behandelte Problem angewandt werden („Flüche der Dimensionalität“, vgl. Abschnitt 2.5.2). Erst die Arbeit von Silver u. a. (2014) versetzte Lillicrap u. a. (2015) in die Lage, insbesondere auch mit überabzählbaren Aktionsräumen umzugehen. Zudem flossen in die vorliegende Dissertation einige Elemente aus Hessel u. a. (2018) wie „verrauschte Netze“ ein, um ein stabileres Lernverhalten zu erzielen.



## 6.3 Umsetzung der flexiblen Planung mit Hilfe neuronaler Netze

Wir möchten nun besprechen, wie die flexible Planung konkret mit Hilfe neuronaler Netze im Rahmen dieser Arbeit umgesetzt wird. Zu diesem Zweck formulieren wir als Vorüberlegung unsere Zielfunktion im Unterabschnitt 6.3.1 um und konstruieren geeignete Repräsentationen der Zustände und Aktionen. Kernstück des flexiblen Planungsansatzes ist das von Lillicrap u. a. (2015) entwickelte Actor-Critic-Verfahren „Deep Deterministic Policy Gradient (DDPG)“, das wir in Unterabschnitt 6.3.2 näher erläutern. Anschließend zeigen wir, wie wir mit Hilfe dieses Verfahrens eine flexible Planung durchführen können.

### 6.3.1 Vorüberlegungen

Um die Bellmansche Rekursionsgleichung anwenden zu können, formulieren wir die Minimum-Funktion in eine Summenfunktion um. Daraufhin stellen wir das verwendete stochastische Modell mit dieser neuen Zielfunktion bezogen auf eine Zeitstufe  $t$  dar. Anschließend besprechen wir die Repräsentation der Zustände und Aktionen.

#### Zielfunktion

Für die Umformung der Zielfunktion hin zu einer Form gemäß Gl. (6.6) verwenden wir das folgende Lemma:<sup>5</sup>

**Lemma 13.** *Für die Systemleistung am Periodenende  $\phi_t \geq 0$ ,  $t = 0, \dots, T$ , gilt*

$$\min_{t=1,\dots,T} \phi_t = \phi_0 + (\psi_1 - \psi_0) + \sum_{t=1,\dots,T-1} (\psi_{t+1} - \psi_t)$$

mit  $\psi_0 := \phi_0$ ,  $\psi_1 := \phi_1$  und  $\psi_t := \min \{\phi_t, \psi_{t-1}\}$  für  $t = 2, \dots, T$ .

*Beweis.* Löst man die Teleskopsumme  $\sum_{t=1,\dots,T-1} (\psi_{t+1} - \psi_t) = \psi_T - \psi_1$  auf, so bleibt zu zeigen, dass

$$\min_{t=1,\dots,T} \phi_t = \underbrace{\phi_0 - \psi_0}_{=0, \text{ da } \psi_0 = \phi_0} + \psi_T.$$

---

<sup>5</sup>Um die Notation zu vereinfachen, nutzen wir die Realisation der Perioden-Systemleistung  $\phi_t$  anstelle der Zufallsvariablen  $\tilde{\phi}_t$ .

Da sich das Prinzip

$$\psi_T = \min \{\phi_T, \psi_{T-1}\} = \min \{\phi_T, \min \{\phi_{T-1}, \psi_{T-2}\}\} = \min \{\phi_T, \phi_{T-1}, \psi_{T-2}\}$$

induktiv weiter fortsetzen lässt, erhält man mit  $\psi_1 = \phi_1$  somit

$$\min_{t=1, \dots, T} \phi_t = \min \{\phi_T, \dots, \phi_1\}.$$

Damit ist das Lemma bewiesen. □

Die Anwendung von Lemma 13 auf die Zufallsvariablen der Zielfunktion ergibt

$$\text{Max.}_{\pi} \mathbb{E} \left( \min_{t=1, \dots, T} \tilde{\phi}_t \right) = \phi_0 + \mathbb{E} \left( \tilde{\psi}_1 - \hat{\psi}_0 \right) + \underbrace{\sum_{t=1, \dots, T-1} \mathbb{E} \left( \tilde{\psi}_{t+1} - \tilde{\psi}_t \right)}_{=:\mathbb{E}(\hat{V}_2(\mathbf{s}_2)|\mathbf{s}_1, \mathbf{x}_1)},$$

$\underbrace{\hspace{10em}}_{=:\hat{V}_1(\mathbf{s}_1)}$

wobei  $\hat{V}_t$  als Approximation der Wertfunktion  $V_t$  Teil der Optimierung ist und  $V_t$  das Ergebnis der Optimierung darstellt. Die Konstante  $\phi_0 = \varphi \left( (\hat{p}_i)_{i=1, \dots, n} \right)$ , die sich durch die Bewertung der Systemleistung über die Anfangszustände der Komponenten zu Beginn der Planung berechnet, wird für die Optimierung nicht benötigt und kann daher entfernt werden.

### Modifiziertes Gesamtmodell

Wir formulieren das stochastische Optimierungsmodell aus Abschnitt 6.1 als Modell der stochastischen dynamischen Programmierung, um eine Politik durch die Anwendung von Reinforcement Learning zu erhalten und letztlich eine flexible Planung implementieren zu können.

Damit erhalten wir die Formulierung:

$$\text{Max.}_{\pi} \mathbb{E} \left( \underbrace{\tilde{\psi}_1 - \hat{\psi}_0}_{=r(\mathbf{s}_1, \mathbf{x}_1, \tilde{\mathbf{s}}_2)} + \sum_{t=1, \dots, T-1} \mathbb{E} \left( \underbrace{\tilde{\psi}_{t+1} - \tilde{\psi}_t}_{=r(\tilde{\mathbf{s}}_{t+1}, \mathbf{x}_{t+1}, \tilde{\mathbf{s}}_{t+2})} \right) \right) \quad (6.8)$$

$$\text{u. d. N. } \tilde{\psi}_t = \min \left\{ \tilde{\psi}_{t-1}, \tilde{\phi}_t \right\} \quad (t = 2, \dots, T) \quad (6.9)$$

$$\tilde{\phi}_t = \varphi((\tilde{p}_{it})_{i=1,\dots,n}) \quad (t = 1, \dots, T) \quad (6.10)$$

$$\tilde{p}_{it} = g_i(x_{it}, \tilde{p}_{i,t-1}, \widetilde{\Delta w}_{it}) \quad (i = 1, \dots, n; t = 1, \dots, T) \quad (6.11)$$

$$B_t = B_{t-1} - \sum_{i=1}^n x_{it} \quad (t = 1, \dots, T) \quad (6.12)$$

$$B_t \geq 0 \quad (t = 1, \dots, T)$$

$$x_{it} = \pi_t(\tilde{p}_{i,t-1}, B_{t-1}) \quad (i = 1, \dots, n; t = 1, \dots, T) \quad (6.13)$$

$$\tilde{\psi}_1 = \tilde{\phi}_1$$

$$\hat{\psi}_0 = \hat{\phi}_0$$

$$\tilde{p}_{i0} = \hat{p}_i \quad (i = 1, \dots, n)$$

$$B_0 = \hat{B}$$

mit Zustand  $\tilde{\mathbf{s}}_t = (\tilde{\mathbf{p}}_{t-1}, B_{t-1})$ , Aktionen  $\mathbf{x}_t = (x_{it})_{i=1,\dots,n}$  sowie einer *Transformationsfunktion*, die sich aus den Gl. (6.9) bis (6.12) zusammensetzt. Auf einer Stufe  $t$  geht man davon aus, dass Stufe  $t-1$  realisiert wurde, sodass in Gl. (6.9) und Gl. (6.11) die Größen  $\tilde{\psi}_{t-1}$  und  $\tilde{p}_{i,t-1}$  als Parameter in ein zeitstufenbezogenes Modell zum Zeitpunkt  $t$  eingehen. Dadurch sind auch die Werte  $x_{it}$ , die über die Politik  $\pi(\tilde{p}_{i,t-1}, B_{t-1})$  in Gl. (6.13) berechnet werden, zum Zeitpunkt  $t$  eindeutig bestimmt. Wie an der Zielfunktion (6.8) zu erkennen ist, ist die Belohnungsfunktion  $r$  unter anderem abhängig vom Folgezustand  $\tilde{\mathbf{s}}_{t+1}$ , d. h., die Belohnung wird erst nach Realisation der Periodenverschleiß  $\widetilde{\Delta w}_{it}$  berechnet. Die Belohnung stellt somit die Anhebung des bisherigen Minimums am Ende der jeweiligen Perioden durch die geleistete Investition dar.

### 6.3.2 Deep Reinforcement Learning: Deep Deterministic Policy Gradient (DDPG)-Algorithmus

Wir stellen nun das Actor-Critic-Verfahren vor, das als erstes die Anwendung von Reinforcement Learning für überabzählbare Aktions- und Zustandsräume erlaubt hat und aufgrund dieser Eigenschaft auch im Rahmen der vorliegenden Arbeit Verwendung findet. Zuvor möchten wir vorstellen, in welcher Form die Zustände und Aktionen in den Algorithmus eingehen werden.

## Repräsentation der Zustände

Den erweiterten Zustand  $\mathring{\mathbf{s}}$  zu Beginn einer Periode  $t$  setzen wir zusammen aus den folgenden Elementen:

$$\mathring{\mathbf{s}}_t = \begin{pmatrix} (\hat{p}_{i,t-1})_{i=1,\dots,n} \\ (\widehat{\Delta w}_{it})_{i=1,\dots,n} \\ \hat{\psi}_{t-1} \\ \hat{\phi}_{t-1} \\ \hat{t} \\ \hat{B}_{t-1} \end{pmatrix}.$$

Das Dach über den Bezeichnern soll jeweils ausdrücken, dass diese Größen zu Beginn der jeweiligen Periode  $t$  im Verlauf des Trainings bekannt sind. Dies sind 1) die prozentualen Abnutzungsvorräte  $\hat{p}_{i,t-1}$  für alle Komponenten  $i$  zu Beginn der Periode  $t$ , 2) die *erwarteten* Periodenverschleiße der einzelnen Komponenten  $\widehat{\Delta w}_{it}$ , 3) das bisherige Minimum der Perioden-Systemleistungen bis zum Ende der Vorperiode  $\hat{\psi}_{t-1} = \min \{\hat{\phi}_{t-1}, \hat{\psi}_{t-2}\}$ , 4) die Perioden-Systemleistung der Vorperiode  $\hat{\phi}_{t-1}$ , 5) der Index der aktuellen Periode  $\hat{t}$  und 6) das Budget  $\hat{B}_{t-1}$  zu Beginn der Periode  $t$ .

Wir nehmen verhältnismäßig viele Elemente in den Zustand auf, damit das neuronale Netz eine größere Menge an Informationen hat, um daraus die Politik  $\pi(\mathring{\mathbf{s}}_t) = \mathbf{x}_t$  abzuleiten. Da die Politik in der modifizierten Zielfunktion die Summe der erwarteten Belohnungen maximiert, die u. a. vom Folgezustand abhängen, können insbesondere die erwarteten Periodenverschleiße beim Training des neuronalen Netzes helfen.

## Repräsentation der Aktionen

Zudem betrachten wir eine alternative Repräsentation des Aktionsraums. Aktuell haben wir  $\mathbf{x}_t = (x_{it})_{i=1,\dots,n}$  mit  $\sum_{i=1}^n x_{it} \leq \hat{B}_{t-1}$ . Es darf also in einer Periode nicht mehr ausgegeben werden als Budget zu Beginn der Periode  $t$  bzw. am Ende der Vorperiode  $t-1$  vorhanden ist. Das bedeutet, dass sich der Definitionsbereich von  $\mathbf{x}$  in jeder Periode in Abhängigkeit des Restbudgets ändert, also  $\mathbf{x}_t \in \mathcal{X}(\hat{B}_{t-1})$ . Um dies zu umgehen, sei nun  $\hat{x}_t^B$  der Anteil am Restbudget  $\hat{B}_{t-1}$  der in Periode  $t$  eingesetzt wird und  $\hat{x}_{it}$  der Anteil

an dem Produkt  $\dot{x}_t^B \cdot \hat{B}_{t-1} = \sum_{i=1}^n x_{it}$ , also

$$\dot{x}_{it} = \frac{x_{it}}{\sum_{j=1}^n x_{jt}} \text{ und } \dot{x}_t^B = \frac{\sum_{i=1}^n x_{it}}{\hat{B}_{t-1}}. \quad (6.14)$$

Sei beispielhaft die Anzahl der Komponenten  $n = 2$ , und das Restbudget zu Beginn einer Periode  $t$  sei  $\hat{B}_{t-1} = 3$ . Wenn nun in der Periode  $t$  insgesamt 2 GE in die Instandhaltung investiert werden, so ist  $\dot{x}_t^B = \frac{2}{3}$ . Wenn in die erste Komponente davon  $x_{1t} = 1,5$  GE investiert werden und in die zweite  $x_{2t} = 0,5$  GE, dann ist  $\dot{x}_{1t} = \frac{3}{4}$  und  $\dot{x}_{2t} = \frac{1}{4}$ .

Damit definieren wir also einen erweiterten Aktionsraum

$$\dot{\mathbf{x}}_t := \left( \dot{x}_{1t}, \dots, \dot{x}_{n-1,t}, \dot{x}_t^B \right)^\top, \quad (6.15)$$

mit  $\dot{\mathbf{x}}_t \in [0; 1]^{n+1}$ . Diese Repräsentation hat den großen Vorteil, dass der Definitionsbereich von  $\dot{\mathbf{x}}_t$  auf jeder Zeitstufe  $t$  gleich ist. Dadurch wird beispielsweise die Anwendung einer einheitlichen Aktivierungsfunktion des neuronalen Netzes ermöglicht, die diesen Wertebereich entsprechend abbildet.

## Beschreibung des Deep Deterministic Policy Gradient-Algorithmus

Der Ablauf des Trainings der neuronalen Netze und somit der Politik für die flexible Planung ist in Alg. 6.1 dargestellt. Zunächst werden in Zeile 0 die Gewichtsvektoren der neuronalen Netze Actor  $\pi$  (für die Approximation der Politik) und Critic  $Q$  (für die Approximation des Q-Werts) zufällig initialisiert<sup>6</sup> und jeweils dupliziert, um deren Ziel-Netzwerke ebenfalls zu initialisieren. Ein Replay-Puffer  $\mathcal{R}$ , in dem später Transitions-Tupel  $(\mathbf{s}_t, \mathbf{x}_t, r_t, \mathbf{s}_{t+1})$  mit  $r_t = r(\mathbf{s}_t, \mathbf{x}_t, \mathbf{s}_{t+1})$  abgespeichert werden, kann – falls vorhanden – mit Trainingsdaten im Voraus (teilweise) belegt werden, was allerdings nicht zwingend erforderlich ist.

Die Zeilen 1 bis 13 werden solange iteriert, bis ein Abbruchkriterium erfüllt ist. Jede dieser Iterationen wird dabei als *Episode* bezeichnet. In Zeile 2 wird ein Zufallsprozess initialisiert, den man für die Exploration der Aktionen benötigt. Exploration bedeutet, neue, noch nicht (oft) besuchte Teile einer Umgebung zu „besuchen“ und Neues zu lernen,

<sup>6</sup>Ziel des Trainings neuronaler Netze ist es, die Gewichtsvektoren so anzupassen, dass das Netz eine bestimmte, häufig nichtlineare Funktion approximiert.

---

**Algorithmus 6.1** Deep Deterministic Policy Gradient (DDPG)-Algorithmus

Quelle: Eigene Darstellung in Anlehnung an Lillicrap u. a. (2015)

---

- 0: **Initialisierung:** Critic-Netzwerk  $Q(\mathring{s}, \mathring{x} \mid \theta^Q)$  und Actor  $\pi(\mathring{s} \mid \theta^\pi)$  zufällig mit den Gewichtsvektoren  $\theta^Q$  und  $\theta^\pi$  initialisieren. Ziel-Netzwerke  $Q'$  und  $\pi'$  mit Gewichtsvektoren  $\theta^{Q'} := \theta^Q$  und  $\theta^{\pi'} := \theta^\pi$  initialisieren. „Replay“-Puffer  $\mathcal{R}$  initialisieren und ggf. mit Daten eines Vortrainings belegen.
- 1: **while** Abbruchkriterium nicht erfüllt
- 2:     Initialisiere einen Zufallsprozess  $\mathcal{N}$  zur Exploration von Aktionen
- 3:     Erhalte den anfänglichen Beobachtungsstatus  $\mathring{s}_1$
- 4:     **for** Periode  $t = 1, \dots, T$  **do**
- 5:         Wähle Aktion  $\mathring{x}_t = \pi(\mathring{s}_t \mid \theta^\pi) + \mathcal{N}_t$  entsprechend der aktuellen Politik und dem Explorations-Rauschen
- 6:         Führe Aktion  $\mathring{x}_t$  aus, beobachte Belohnung  $r_t$ , und ziehe zufällig neuen Zustand  $\mathring{s}_{t+1}$
- 7:         Speichere Transition  $(\mathring{s}_t, \mathring{x}_t, r_t, \mathring{s}_{t+1})$  in  $\mathcal{R}$
- 8:         Sample ein Mini-Batch von  $N$  Transitionen  $(\mathring{s}_j, \mathring{x}_j, r_j, \mathring{s}_{j+1})$  von  $\mathcal{R}$
- 9:         Setze  $y_{j+1} = r_j + \gamma \cdot Q'(\mathring{s}_{j+1}, \pi'(\mathring{s}_{j+1} \mid \theta^{\pi'}) \mid \theta^{Q'})$
- 10:         Aktualisiere Critic-Netzwerk  $Q$  durch Minimierung über die Gewichte  $\theta^Q$  der Verlustfunktion  $L = \frac{1}{N} \sum_j (y_{j+1} - Q(\mathring{s}_j, \mathring{x}_j \mid \theta^Q))^2$
- 11:         Aktualisiere Actor-Netzwerk  $\pi$  unter Verwendung des gesampleten Policy-Gradienten:

$$\nabla_{\theta^\pi} J(\theta^\pi) \approx \frac{1}{N} \sum_{j=1}^N \nabla_{\mathring{x}} Q(\mathring{s}, \mathring{x} \mid \theta^Q) \Big|_{\mathring{s}=\mathring{s}_j, \mathring{x}=\pi(\mathring{s}_j)} \nabla_{\theta^\pi} \pi(\mathring{s} \mid \theta^\pi) \Big|_{\mathring{s}_j}$$

$$\theta^\pi \leftarrow \theta^\pi + \alpha \cdot \nabla_{\theta^\pi} J(\theta^\pi)$$

- 12:         Aktualisiere die Zielfunktionen

$$\theta^{Q'} \leftarrow \delta \theta^Q + (1 - \delta) \theta^{Q'}$$

$$\theta^{\pi'} \leftarrow \delta \theta^\pi + (1 - \delta) \theta^{\pi'}$$

- 13:     **end for**
-

Exploitation hingegen, bereits (oft) besuchte Teile der Umgebung zu besuchen und bereits Ausgeführtes zu verbessern.

Jede Episode startet zum Zeitpunkt  $t = 1$  zum Zustand  $\hat{\mathbf{s}}_1$  (Zeile 3) und simuliert dann alle Zeitschritte  $t = 1, \dots, T$  (Zeile 4). Zunächst werden die Aktionen  $\hat{\mathbf{x}}_t$  für alle Komponenten gemäß der aktuellen Politik  $\hat{\mathbf{x}}_t = \pi(\mathbf{s}_t \mid \boldsymbol{\theta}^\pi)$  (zzgl. des Explorationsrauschens, das nur in der Trainingsphase des Netzes angewendet wird) bestimmt. Statt der Addition eines Ornstein-Uhlenbeck-Prozesses<sup>7</sup>, der bei Lillicrap u. a. (2015) vorgeschlagen wird, verwendet die Implementierung des Kapitels 7 einen Beta-Prozess mit Erwartungswert  $\hat{x}_{it}$  sowie Beschränkung auf den Bereich  $[0; 1]$  und über die Episoden abnehmender Varianz. „Aktuelle Politik“ bedeutet, dass das Actor-Netz  $\pi$  gemäß dem Input  $\hat{\mathbf{s}}_t$  und dem aktuellen Gewichtsvektor  $\boldsymbol{\theta}^\pi$  einen Wert  $\hat{\mathbf{x}}_t$  zurückliefert („Output“), also  $\hat{\mathbf{x}}_t = \pi(\mathbf{s}_t \mid \boldsymbol{\theta}^\pi)$ .

Zeile 6 simuliert einen Zeitschritt, indem eine Stichprobe des gammaverteilten Periodenverschleißes gezogen wird, die Belohnung  $r_t$  – in unserem Fall eine mögliche Erhöhung des Systemleistungsminimums – beobachtet und in den Folgezustand  $\mathbf{s}_{t+1}$  übergegangen wird. Diese Transition  $(\mathbf{s}_t, \mathbf{x}_t, r_t, \mathbf{s}_{t+1})$  wird im Replay-Puffer  $\mathcal{R}$  abgespeichert (Zeile 7). Zeile 8 stellt den Schritt des Experience Replay dar. Es wird zufällig eine vorgegebene Anzahl  $N$  an Transitionstupeln entnommen. Das Experience Replay wird vor allem aus zwei Gründen durchgeführt: 1) Es ist eine effizientere Nutzung früherer Erfahrungen möglich, indem mehrfach damit gelernt wird, und 2) es wird i. A. ein besseres Konvergenzverhalten beim Training erreicht, da die Daten (bei genügend großem Replay-Puffer) dadurch annähernd unabhängig und identisch verteilt sind (dies wird in den meisten Konvergenzbeweisen des *überwachten* Lernens<sup>8</sup> angenommen).

In Zeile 9 wird für jedes der in Zeile 8 zufällig entnommenen Transitionstupel der Wert des Folgezustands  $y_{j+1} \approx \hat{V}^\pi(\mathbf{s}_{j+1})$  über die Ziel-Netzwerke des Critic und des Actor berechnet. Die Approximation gilt aufgrund der Beziehung (6.7) auf S. 78. Dabei wird die Politik  $\pi$  über das Ziel-Netzwerk des Actor angenähert, das die Aktion  $\mathbf{x}_{j+1}$  für die Berechnung des Q-Werts liefert, die wiederum über das Ziel-Netzwerk des Critic erfolgt. Anschließend werden in Zeile 10 die Gewichte des Critic (nicht dessen Zielnetzwerk) gemäß der Minimierung einer Verlustfunktion aktualisiert. Die Verlustfunktion misst den

---

<sup>7</sup>Dieser Prozess macht vor allem dann Sinn, wenn die Entscheidungsvariablen ein Bewegungsmomentum mit berücksichtigen sollen.

<sup>8</sup>Zum Begriff des überwachten Lernens vgl. z. B. Russell und Norvig (2016).

mittleren quadrierten Abstand zwischen den berechneten Werten  $y_{j+1}$  und den zugehörigen Q-Werten in Abhängigkeit von  $\mathbf{s}_j$  und  $\mathbf{x}_j$ . Hier wird deutlich, dass der Zustand  $\mathbf{s}_j$  genügend Informationen – in unserem Fall also über den Zeitpunkt, das Restbudget usw. – enthalten muss, damit das neuronale Critic-Netz eine gute Funktionsapproximation liefern kann (vgl. Abschnitt „Repräsentation der Zustände“). Analog muss der Aktionsraum immer im gleichen Definitionsbereich liegen, damit das Actor-Netz sinnvolle Werte zurückgeben kann (vgl. Abschnitt „Repräsentation der Aktionen“).

Zeile 11 beschreibt nun den namensgebenden deterministischen Politik-Gradienten (vgl. Silver u. a. (2014)). Hier werden die Gewichte des Actor-Netzwerks aktualisiert, und zwar über den Gradientenaufstieg

$$\boldsymbol{\theta}^\pi \leftarrow \boldsymbol{\theta}^\pi + \alpha \cdot \nabla_{\boldsymbol{\theta}^\pi} J(\boldsymbol{\theta}^\pi)$$

mit Schrittweite  $\alpha$ . Die Funktion  $J(\boldsymbol{\theta}^\mu)$ , deren Gradient berechnet wird, ist dabei eine Schätzung des erwarteten Q-Werts  $\mathbb{E}\left(Q\left(\mathbf{s}, \pi(\mathbf{s} \mid \boldsymbol{\theta}^\pi) \mid \boldsymbol{\theta}^Q\right)\right)$ , der über die Kettenregel nach den Gewichtsvektoren des Actor abgeleitet wird.

In Zeile 12 werden zuletzt die Gewichtsvektoren der Zielnetzwerke über eine exponentielle Glättung erster Ordnung mit einem Glättungsparameter  $0 < \delta < 1$  aktualisiert, indem eine Konvexkombination zwischen den vorherigen Gewichten der Zielnetzwerke und den aktualisierten Gewichten von Actor und Critic gebildet werden. Im Anschluss wird der nächste Zeitschritt oder am Ende des Planungszeitraums die nächste Episode gestartet. Die Verwendung der Ziel-Netzwerke soll dazu führen, ein stabileres Training zu ermöglichen.

Nachdem das Training abgeschlossen ist, hat man als Ergebnis u. a. einen Gewichtsvektor  $(\boldsymbol{\theta}^\pi)^*$  für den Actor erhalten. Damit hat man auch direkt eine Politik  $\pi(\mathbf{s}_t \mid (\boldsymbol{\theta}^\pi)^*)$  erhalten, die auf jeder Zeitstufe  $t$  und für jeden Zustand  $\mathbf{s}_t$  im Rahmen der flexiblen Planung eine Empfehlung für eine Investition  $(x_{it})_{i=1,\dots,n}$  in die Instandhaltung aller Komponenten  $i$  über Gl. (6.14) liefert mit

$$\left((\dot{x}_{it})_{i=1,\dots,n}^\top, \dot{x}_t^B\right)^\top = \dot{\mathbf{x}}_t = \pi(\mathbf{s}_t \mid (\boldsymbol{\theta}^\pi)^*),$$

also

$$x_{it} = \dot{x}_{it} \cdot \dot{x}_t^B \cdot \hat{B}_{t-1}.$$



## Anwendbarkeit des Algorithmus

Damit der Algorithmus angewendet werden kann, sollte ein Markovscher Entscheidungsprozess vorliegen. Dadurch, dass die Periodenverschleiße annahmegemäß unabhängig von der Verschleißhistorie sind, ist diese Bedingung bei uns erfüllt. Konvergenzbeweise konnten in der Literatur nur im Bereich des überwachten Lernens geführt werden, bei dem Trainingsdaten schon vorhanden sind, und nicht im Bereich des Reinforcement Learning, wo die Trainingsdaten erst im Zeitverlauf generiert werden (vgl. Richter (2019, S. 255 ff.) oder Watkins und Dayan (1992)). Es wird dafür die notwendige Annahme getroffen, dass die generierten Trainingsdaten des Reinforcement Learning identisch und unabhängig voneinander verteilt sind. Mit der Anwendung von Experience Replay wird versucht, diese Annahme annähernd zu erfüllen.

Üblicherweise wird beim Reinforcement Learning zudem ein unendlicher Planungszeitraum angenommen. Dadurch, dass man aber potenziell unendlich viele Episoden von endlichen Planungszeiträumen durchlaufen kann und durch das Experience Replay eine rein zufällige Auswahl aus dem Replay-Puffer entnommen wird, lassen sich die Überlegungen auch auf unseren Fall übertragen.

# 7

## Kapitel 7

# Vergleichende experimentelle Performance-Analysen

Wir führen nun Performance-Analysen auf der Grundlage von konkreten Instanzen durch, die wir in Abschnitt 7.1 zunächst vorstellen. Mit diesen Instanzen werden wir zum einen zeigen, dass eine (starre) periodenübergreifende Planung Vorteile gegenüber einer (starren) periodenbezogenen Planung hat sowie weitere Analysen für die starre Planung durchführen (Abschnitt 7.2) und im Anschluss die vorgestellten Planungsansätze rollierende und flexible Planung mit DDPG untereinander und gegenüber einer zufälligen Budgetallokation vergleichen (Abschnitt 7.3).

## 7.1 Konfiguration der Instanzen

Wir stellen nun vor, wie die verschiedenen Instanzen, die wir betrachten, konfiguriert wurden. Wir strukturieren die Konfiguration nach den Bereichen 1) Strukturbewertungsfunktion, 2) Zustandsübergangsfunktion, 3) Verschleiß und 4) Anfangswertbedingungen.

### 7.1.1 Strukturbewertungsfunktionen

Als Strukturen des Systems wählen wir fünf unterschiedliche Konfigurationen:

1. Seriensystem mit fünf Komponenten
2. Parallelsystem mit fünf Komponenten
3. Serien-Parallel-System mit fünf Komponenten
4. Brückenschaltung mit fünf Komponenten
5. Brückenschaltung mit acht Komponenten

Für Parallelsysteme definieren wir

$$\varphi^{\text{Par}}(\mathbf{p}) = 1 - \prod_{i=1, \dots, n} (1 - p_i).$$

Die Strukturbewertungsfunktionen für die Serien- bzw. Parallelschaltung mit fünf Komponenten  $i \in \{1, \dots, 5\}$  lauten

$$\varphi^{\text{Ser}}(\mathbf{p}) = \prod_{i=1}^5 p_i$$

bzw.

$$\varphi^{\text{Par}}(\mathbf{p}) = 1 - \prod_{i=1}^5 (1 - p_i).$$

Die Serien-Parallel-Schaltung (s. Abb. 7.1) hat die Strukturbewertungsfunktion

$$\varphi^{\text{SerPar}}(\mathbf{p}) = 1 - (1 - p_1 \cdot p_2 \cdot p_3) \cdot (1 - p_4 \cdot p_5).$$

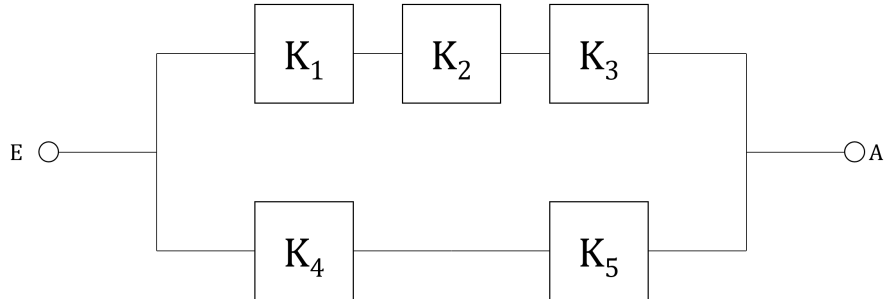


Abb. 7.1: Strukturbewertungsfunktion der Serien-Parallel-Schaltung

Quelle: Eigene Darstellung in Anlehnung an Beichelt (1993, S. 63)

Die Brückenschaltung mit fünf Komponenten  $i \in \{1, \dots, 5\}$  (s. Abb. 7.2) besitzt die folgende Strukturbewertungsfunktion, die wir mit der Pivotzerlegung aus Beichelt und Tittmann (2012, S. 27 ff.) berechnet haben:

$$\begin{aligned} \varphi^{\text{Brücke}}(\mathbf{p}) = & p_5 \cdot (1 - (1 - p_1) \cdot (1 - p_2)) \cdot (1 - (1 - p_3) \cdot (1 - p_4)) \\ & + (1 - p_5) \cdot (1 - (1 - p_1 \cdot p_3) \cdot (1 - p_2 \cdot p_4)). \end{aligned}$$

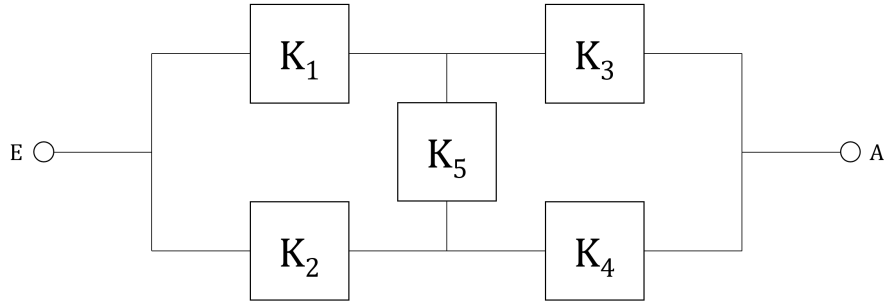


Abb. 7.2: Strukturbewertungsfunktion der Brückenschaltung mit fünf Komponenten  
Quelle: Eigene Darstellung in Anlehnung an Beichelt (1993, S. 63)

Die Brückenschaltung (s. Abb. 7.3 auf der nächsten Seite) mit acht Komponenten  $i \in \{1, \dots, 8\}$  wurde ebenfalls über die Pivotzerlegung berechnet. Die Strukturbewertungsfunktion für die achtkomponentige Brückenschaltung lautet somit

$$\begin{aligned} \varphi^{\text{Brücke8}}(\mathbf{p}) = & p_5 \cdot p_8 \cdot S_{5,8} + (1 - p_5) \cdot p_8 \cdot S_{\bar{5},8} \\ & + p_5 \cdot (1 - p_8) \cdot S_{5,\bar{8}} + (1 - p_5) \cdot (1 - p_8) \cdot S_{\bar{5},\bar{8}} \end{aligned}$$

mit

$$\begin{aligned} S_{5,8} &= (1 - (1 - p_1) \cdot (1 - p_2) \cdot (1 - p_6)) \cdot (1 - (1 - p_3) \cdot (1 - p_4) \cdot (1 - p_7)), \\ S_{\bar{5},8} &= 1 - (1 - (1 - (1 - p_6) \cdot (1 - p_2)) \cdot (1 - (1 - p_7) \cdot (1 - p_4))) \cdot (1 - p_1 \cdot p_3), \\ S_{5,\bar{8}} &= 1 - (1 - (1 - (1 - p_1) \cdot (1 - p_2)) \cdot (1 - (1 - p_3) \cdot (1 - p_4))) \cdot (1 - p_6 \cdot p_7), \\ S_{\bar{5},\bar{8}} &= 1 - (1 - p_1 \cdot p_3) \cdot (1 - p_2 \cdot p_4) \cdot (1 - p_6 \cdot p_7). \end{aligned}$$

Für die Performance-Analyse wurden für jede der fünf Strukturen jeweils fünf Instanzen mit Parameterwerten erzeugt, die gleichverteilt aus gegebenen Intervallen gezogen wurden. Die konkreten Intervalle finden sich in den zugehörigen folgenden Abschnitten.

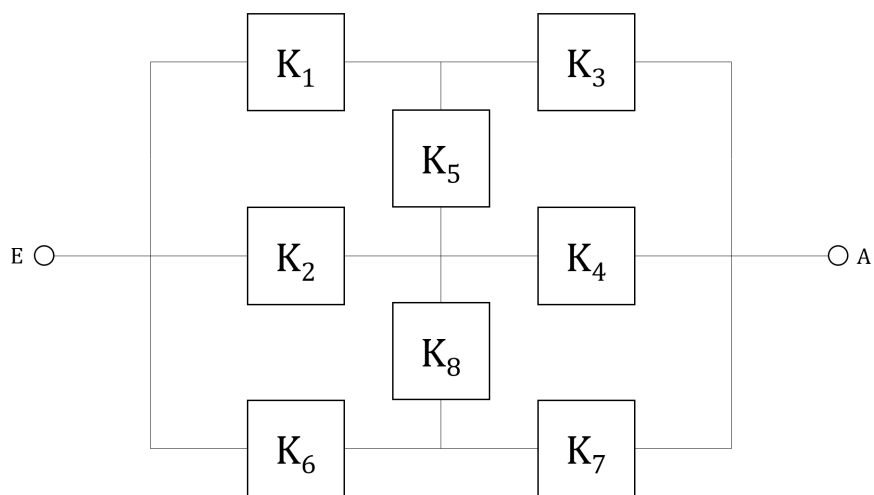


Abb. 7.3: Strukturbewertungsfunktion der Brückenschaltung mit acht Komponenten  
Quelle: Eigene Darstellung in Anlehnung an Beichelt (1993, S. 63)

### 7.1.2 Zustandsübergangsfunktion

Als Zustandsübergangsfunktion wollen wir diejenige verwenden, die wir in Abschnitt 5.1.2 hergeleitet haben, also<sup>1</sup>

$$g_i(x_{it}, p_{i,t-1}, \widehat{\Delta w}_{it}) = p_{it} = \max \left\{ 1 - (1 - p_{i,t-1}) \cdot \exp(-\widehat{a}_i \cdot x_{it}) - \widehat{\Delta w}_{it}; 0 \right\}$$

mit Skalierungsparametern  $\widehat{a}_i > 0$ . Für diesen gilt: je höher der Skalierungsparameter einer Komponente ist, desto mehr bewirkt der Einsatz finanzieller Mittel pro GE bezüglich des prozentualen Abnutzungsvorrats dieser Komponente. Dieser Parameter kann in der Praxis aus vorhandenen historischen Kosten geschätzt werden.

Die konkreten Werte für die gerechneten Instanzen wurden mit Python generiert (vgl. Anhang B.2.2) mit  $\widehat{a}_i^\varphi \sim \mathcal{U}[0, 1; 5]$  für alle  $i, \varphi$  und sind auf eine Nachkommastelle gerundet in Tab. B.1 auf Seite 163 im Anhang angegeben.<sup>2</sup> Der hochgestellte Index  $\varphi$  gibt hier und im weiteren Verlauf jeweils die zugehörige Struktur an, also  $\varphi \in \{\text{Ser}, \text{Par}, \text{SerPar}, \text{Brücke}, \text{Brücke8}\}$  gemäß der Bezeichner aus dem vorangehenden Abschnitt.

### 7.1.3 Verschleißparametrisierung

In Abschnitt 4.2 hatten wir hergeleitet, dass man sinnvolle Werte für die Parameter  $\widehat{\Delta w}_{it}$  mit

$$\widehat{\Delta w}_{it} := \mathbb{E}(\widetilde{W}(t+1)) - \mathbb{E}(\widetilde{W}(t))$$

erhält, wobei  $\widetilde{W}(t)$  gammaverteilt ist. Der Erwartungswert des kumulierten Verschleißes

$$\mathbb{E}(\widetilde{W}(t)) = \frac{v(t)}{u}$$

---

<sup>1</sup>Für die rollierende Planung, die wir mit CONOPT/GAMS über eine Python-Schnittstelle implementieren, approximieren wir die Maximum-Funktion gemäß Drud (2021b).

<sup>2</sup> $\mathcal{U}[a; b]$  stellt die Gleichverteilung im Intervall zwischen  $a$  und  $b$  dar. Gerechnet wurde für alle Parameter mit nicht-gerundeten Zahlen.

kann für die Parametrisierung gemäß van Noortwijk (2009) angenähert werden mit

$$\frac{v(t)}{\hat{u}} \approx \frac{\hat{c} \cdot t^{\hat{b}}}{\hat{u}}.$$

Ausgehend von den bekannten Erwartungswerten werden wir also gammaverteilte Zufallsvariablen parametrisieren und letztlich simulieren. Konkret werden wir für die Parametrisierung die Werte aus den Tab. B.2 auf Seite 164 und B.3 auf Seite 165 des Anhangs in unseren Rechenbeispielen verwenden. Für  $\hat{b}_i^\varphi > 1$  steigt der Verschleiß jede Periode etwas stärker an, wodurch wir einen nichtstationären Prozess simulieren.

Es gilt i. A.  $\hat{u}_i^\varphi \sim \mathcal{U}[15; 20]$ ,  $\hat{b}_i^\varphi \sim \mathcal{U}[1; 1,5]$  und  $\hat{c}_i^\varphi \sim \mathcal{U}[0,6; 1,2]$  alle  $i, \varphi$ . Nur für das Parallelsystem wurde die Ausnahme  $\hat{u}_i^\varphi = \hat{u}_i^{\text{Par}} \sim \mathcal{U}[60; 80]$  gemacht, um sinnvollere Instanzen zu erhalten.<sup>3</sup>

#### 7.1.4 Anfangswertbedingungen

Für die mehrperiodigen Instanzen gehen wir von einem Planungszeitraum von fünf Perioden aus ( $T = 5$ ) mit einem Budget von  $\hat{B} = 5$ . Die Anfangswertbedingungen für die prozentualen Abnutzungsvorräte finden sich in Tab. B.4 auf Seite 166 im Anhang.

Es gilt abweichend für das Parallelsystem  $\hat{p}_i^\varphi = \hat{p}_i^{\text{Par}} \sim \mathcal{U}[0,1; 0,2]$ , um aussagekräftigere Instanzen zu erzeugen, deren Zielfunktionswerte nicht alle der Obergrenze 1 entsprechen, und sonst  $\hat{p}_i^\varphi \sim \mathcal{U}[0,7; 0,9]$  für alle weiteren  $i, \varphi$ .

## 7.2 Vergleiche für die starre Planung mit dem Erwartungswertmodell

Wir möchten nun insbesondere den periodenübergreifenden Ansatz mit einer Planung vergleichen, für die das Budget in jeder Periode einer Budgetrestriktion unterliegt. Im Anschluss gehen wir auf Fragestellungen ein, die sich im Verlauf der Analyse des Erwartungswertmodells in Kapitel 5 ergeben haben, nämlich wie gut die Grenznutzenmethode

---

<sup>3</sup>Die Verschleißparameter wurden insbesondere so gewählt, dass im Erwartungswert für die optimale Lösung alle Abnutzungsvorräte  $p_{it}^*$  streng positiv sind.

im Vergleich zu den exakten Solvern LindoGlobal und BARON das einperiodige Erwartungswertmodell löst und wie zuverlässig der Solver CONOPT3 im Vergleich mit LindoGlobal und BARON optimale Ergebnisse für das mehrperiodige Erwartungswertmodell liefert.

### 7.2.1 Vergleich periodenbezogener und periodenübergreifender Budgetallokation

Mit den im vorangehenden Abschnitt 7.1 beschriebenen Instanzen führen wir nun eine erste vergleichende Performance-Analyse mit Hilfe der Python-Schnittstelle von GAMS mit CONOPT durch. Dabei vergleichen wir die Zielfunktionswerte einer starren Planung vom Erwartungswertmodell mit 1) periodenübergreifender Budgetallokation und 2) mit periodenbezogener Budgetallokation. Bei der zweiten Variante wird das Budget  $\hat{B}$  gleichmäßig auf die Perioden  $t = 1, \dots, T$  verteilt, sodass für jede Periode finanzielle Mittel in Höhe von  $\frac{\hat{B}}{T} = \frac{5}{5} = 1$  zur Verfügung stehen. Dies wird durch das Hinzufügen einer Nebenbedingung zum Erwartungswertmodell Gl. (5.29) – (5.31) erreicht, nämlich

$$B_t = \hat{B} - \frac{\hat{B}}{T} \cdot t \quad (t = 1, \dots, T).$$

Das Restbudget  $B_t$  am Ende der Periode  $t$  nimmt also in jeder Periode genau  $\frac{\hat{B}}{T}$  ab, sodass  $B_T = 0$  gilt. Die Ergebnisse der experimentellen Analyse sind in Tab. 7.1 auf Seite 97 zusammengefasst. Man sieht dort für verschiedene Systeme die einzelnen optimalen Zielfunktionswerte  $z$  und den relativen Vorteil  $\frac{z^{\hat{u}} - z^b}{z^b}$  der periodenübergreifenden (mit Zielfunktionswert  $z^{\hat{u}}$ ) gegenüber der periodenbezogenen (mit Zielfunktionswert  $z^b$ ) Budgetallokation.

Man kann der Tabelle entnehmen, dass die Höhe des relativen Vorteils stark unterschiedlich ist und von den einzelnen Systemkonfigurationen abhängt. Bei den Parallelsystemen kann man bis auf die vierte Nachkommastelle beispielsweise keinen Unterschied zwischen periodenübergreifender und periodenbezogener Budgetallokation erkennen. Da sich der Wertebereich der Zielfunktionswerte schon für die periodenbezogene Budgetallokation knapp unter dem maximal möglichen Wert 1 bewegt, ist weiterer Optimierungsspielraum für die periodenübergreifende Budgetallokation hier kaum vorhanden.



In anderen Fällen, in denen die Zielfunktionen deutlich geringere Werte im Optimum besitzen, zeichnet sich ein deutliches Bild ab. Insbesondere beim Seriensystem kann z. B. der relative Vorteil mehr als ein Drittel betragen (Seriensystem, Instanz Nr. 5). Über alle betrachteten Instanzen zusammen ergibt sich im Mittel ein relativer Vorteil der periodenübergreifenden Budgetallokation gegenüber der periodenbezogenen von 5,55 %.

Es zeigt sich somit anhand dieser experimentellen Analyse, dass es sinnvoll ist, in mehrperiodigen Modellen der Instandhaltung modularer Systeme immer wenn möglich eine periodenübergreifende und keine periodenbezogene Budgetallokation durchzuführen. Somit bringt der Ansatz der periodenübergreifenden Budgetallokation auch einen Fortschritt gegenüber den Ansätzen aus der Literatur, die eine periodenbezogene Budgetallokation annehmen.

### 7.2.2 Weitere Analysen

Wir möchten nun die Ergebnisse zu den Untersuchungen der Grenznutzenmethode für das einperiodige Problem und des Solververgleichs für das mehrperiodige Problem besprechen.

#### Vergleich der Grenznutzenmethode mit exakten Standard-Solvern für das einperiodige Modell

Wir vergleichen die Grenznutzenmethode, die in Abschnitt 5.3 entwickelt wurde, sowohl mit dem exakten Solver LindoGlobal als auch mit dem exakten Solver BARON, und zwar hinsichtlich Rechendauer, Anzahl Iterationen und Zielfunktionswert. Dafür wurden nicht die zufallsgenerierten Instanzen aus Abschnitt 7.1 betrachtet, sondern die Konfiguration  $\hat{B} = 1$ ,  $n = 5$ ,  $\hat{\mathbf{p}} = (0,5; 0,6; 0,7; 0,8; 0,6)^\top$ ,  $\hat{\mathbf{a}} = (2; 4; 3; 2; 1)^\top$ ,  $\widehat{\Delta\mathbf{w}} = (0,15; 0,1; 0,2; 0,3; 0,2)^\top$  und, da es sich um eine einperiodige Betrachtung handelt,  $T = 1$ . Dies wurde für alle fünfkomponentigen Strukturbewertungsfunktionen (vgl. Abschnitt 7.1.1) durchgeführt. Für die Grenznutzenmethode wurde als Abbruchbedingung  $\max\{v, R\} < \varepsilon$  gewählt mit Schrittweite  $v$  und Spannweite der relevanten Grenznutzen  $R = \max_{i \in \{i | x_i > 0\}} \gamma_i - \min_{i \in \{i | x_i > 0\}} \gamma_i$ .

Die Ergebnisse sind in Tab. 7.2 auf Seite 98 zusammengefasst. Dabei wurden als relative und absolute Optimalitätstoleranzen jeweils 0 gefordert, wobei diese intern von

System	Seriell		
Instanznr.	$z^{\ddot{u}}$	$z^b$	$\frac{z^{\ddot{u}}-z^b}{z^b}$
1	0,1889	0,1450	30,23%
2	0,2351	0,1852	26,89%
3	0,3368	0,3159	6,62%
4	0,2491	0,2393	4,08%
5	0,1967	0,1473	33,57%
Mittelwert	0,2413	0,2066	20,28%

(a) Seriensystem

System	Parallel		
Instanznr.	$z^{\ddot{u}}$	$z^b$	$\frac{z^{\ddot{u}}-z^b}{z^b}$
1	1,0000	1,0000	0,00%
2	1,0000	1,0000	0,00%
3	0,9996	0,9996	0,00%
4	0,9984	0,9984	0,00%
5	0,9999	0,9999	0,00%
Mittelwert	0,9996	0,9996	0,00%

(b) Parallelsystem

System	Serien-Parallel		
Instanznr.	$z^{\ddot{u}}$	$z^b$	$\frac{z^{\ddot{u}}-z^b}{z^b}$
1	0,6774	0,6508	4,09%
2	0,6986	0,6568	6,35%
3	0,8381	0,8350	0,38%
4	0,8103	0,7774	4,23%
5	0,8042	0,7869	2,20%
Mittelwert	0,7657	0,7414	3,45%

(c) Serien-Parallel-System

System	Brücke, 5 Komp.		
Instanznr.	$z^{\ddot{u}}$	$z^b$	$\frac{z^{\ddot{u}}-z^b}{z^b}$
1	0,8712	0,8325	4,65%
2	0,9392	0,9356	0,38%
3	0,7833	0,7457	5,04%
4	0,8516	0,8342	2,09%
5	0,8947	0,8856	1,03%
Mittelwert	0,8680	0,8467	2,64%

(d) Brückenschaltung (5 Komp.)

System	Brücke, 8 Komp.		
Instanznr.	$z^{\ddot{u}}$	$z^b$	$\frac{z^{\ddot{u}}-z^b}{z^b}$
1	0,9697	0,9643	0,56%
2	0,9394	0,9241	1,66%
3	0,9759	0,9712	0,48%
4	0,9319	0,9025	3,25%
5	0,9690	0,9585	1,09%
Mittelwert	0,9572	0,9441	1,41%

(e) Brückenschaltung (8 Komp.)

Tab. 7.1: Ergebnisse des Vergleichs der optimalen Zielfunktionswerte periodenübergreifender und periodenbezogener Budgetallokation, gerundet auf vier Nachkommastellen

Quelle: Eigene Darstellung

## 7 Vergleichende experimentelle Performance-Analysen

Lösungsmethode	Zfw. (Python)	Zfw. (GAMS)	# Iter.	Rechendauer in Sek.
Grenznutzenmethode, $\varepsilon = 10^{-4}$	0,0681072977811	–	10.003	3,3
Grenznutzenmethode, $\varepsilon = 10^{-5}$	0,0682178670029	–	100.002	21,7
BARON	0,0682178661035	0,0682237485752	5	< 1
LindoGlobal	0,0682178670138	0,0682178758898	222.024	100*

(a) Seriensystem

Lösungsmethode	Zfw. (Python)	Zfw. (GAMS)	# Iter.	Rechendauer in Sek.
Grenznutzenmethode, $\varepsilon = 10^{-4}$	0,9928557227347	–	10.000	4,2
Grenznutzenmethode, $\varepsilon = 10^{-5}$	0,9928557227472	–	100.000	43,5
BARON	0,9928542505272	0,9928615350395	107	< 1
LindoGlobal	0,9928557227473	0,9928557227473	553.020	100*

(b) Parallelsystem

Lösungsmethode	Zfw. (Python)	Zfw. (GAMS)	# Iter.	Rechendauer in Sek.
Grenznutzenmethode, $\varepsilon = 10^{-4}$	0,4655108562674	–	10.015	3,8
Grenznutzenmethode, $\varepsilon = 10^{-5}$	0,4655108578525	–	100.020	48,6
BARON	0,4655031364353	0,4655141538921	33	< 1
LindoGlobal	0,4655108578919	0,4655109570595	15.905	4

(c) Serien-Parallel-System

Lösungsmethode	Zfw. (Python)	Zfw. (GAMS)	# Iter.	Rechendauer in Sek.
Grenznutzenmethode, $\varepsilon = 10^{-4}$	0,6967606127223	–	10.007	5,2
Grenznutzenmethode, $\varepsilon = 10^{-5}$	0,6970939374923	–	100.001	51,7
BARON	0,6970878908237	0,6971023282647	315	< 1
LindoGlobal	0,6970939375015	0,6970939622182	875.078	100*

(d) Brückenschaltung (5 Komp.)

Tab. 7.2: Ergebnisse des Vergleichs der Grenznutzenmethode mit exakten Solvern BARON und LindoGlobal, gerundet auf 13 Nachkommastellen

Quelle: Eigene Darstellung

GAMS/BARON auf  $10^{-9}$  angehoben werden.<sup>4</sup> Rechendauern, die mit \* markiert sind, deuten darauf hin, dass das gesetzte Zeitlimit von 100 Sekunden<sup>5</sup> erreicht wurde, ohne eine (nach Ausgabe von GAMS) optimale Lösung zurückzuliefern. Da sich die Zielfunktionswerte interessanterweise unterscheiden, wenn sie durch GAMS oder Python evaluiert wurden, sind beide Werte für BARON und LindoGlobal aufgeführt.

**Rechendauer.** Die Rechendauer bei der Grenznutzenmethode ist abhängig von der Genauigkeit der Lösung, die mit dem Parameter  $\varepsilon$  gesteuert wird. Für  $\varepsilon = 10^{-4}$  dauerten die Durchläufe weniger als 10 Sekunden, für  $\varepsilon = 10^{-5}$  immer noch weniger als eine Minute. Das Verhalten von BARON und LindoGlobal zeigte sich unterschiedlich. Für alle Strukturen außer dem Serien-Parallel-System fand LindoGlobal lt. GAMS keine optimale Lösung innerhalb von 100 Sekunden. BARON lieferte stets eine angeblich optimale Lösung in unter einer Sekunde zurück. Für das Serien-Parallel-System fand auch LindoGlobal eine lt. GAMS optimale Lösung innerhalb von vier Sekunden.

**Anzahl Iterationen.** Während bei der Grenznutzenmethode in etwa  $\frac{1}{\varepsilon}$  Iterationen durchlaufen werden, bis das Abbruchkriterium erreicht ist, ist dies bei den exakten Solvern von Instanz zu Instanz unterschiedlich. BARON findet eine Lösung innerhalb von 5 bis 315 Iterationen, während LindoGlobal im Fall des Seriensystems 15.905 Iterationen durchführt, ansonsten bis zu 875.078 Iterationen benötigt.

**Zielfunktionswert.** Auf den ersten Blick scheint BARON beim Vergleich der unterschiedlichen Verfahren die anderen hinsichtlich der gewählten Kriterien, aber insbesondere bezogen auf den von GAMS errechneten Zielfunktionswert, zu dominieren. Interessanterweise weichen die Zielfunktionswerte, die von GAMS ermittelt wurden, insbesondere beim Solver BARON stark von einer Evaluation durch Python nach oben ab. So haben die Zielfunktionswerte von BARON in GAMS z. B. bei der Brückenschaltung schon in der vierten Nachkommastelle Abweichungen nach oben. Der mit Python auf Basis der Lösung  $(x_i^*)_{i=1,\dots,5}$  ermittelte Funktionswert liegt allerdings unter denen der Grenznutzenmethode mit  $\varepsilon = 10^{-5}$  und LindoGlobal. Daher kann festgehalten werden, dass für die vorliegende Problemstellung BARON nicht zuverlässig genug arbeitet. Im Vergleich mit LindoGlobal sind die Werte der Grenznutzenmethode mit  $\varepsilon = 10^{-4}$  deutlich schlechter, für  $\varepsilon = 10^{-5}$  sind die Zielfunktionswerte in Python vergleichbar.

Es lässt sich zusammenfassend sagen, dass BARON zwar mit Abstand die kürzesten

---

<sup>4</sup>Vgl. z. B. GAMS-Code im Abschnitt B.2.1, Options-Parameter *optcr* und *optca*.

<sup>5</sup>Es wurden vereinzelt Tests mit 1000 Sekunden durchgeführt, die zum gleichen Ergebnis führten.

Rechenzeiten benötigt, dafür aber auch nicht hinreichend genaue Ergebnisse liefert. Die Grenznutzenmethode ist für ein Abbruchkriterium von  $\varepsilon = 10^{-4}$  zu ungenau. Für  $\varepsilon = 10^{-5}$  bietet sie ebenso zuverlässige Lösungen wie der Solver LindoGlobal, welcher allerdings für die betrachteten Beispiel nur für das Serien-Parallel-System innerhalb des gesetzten Zeitlimits eine eindeutige Aussage hinsichtlich der Optimalität liefert.

### **Vergleich von CONOPT3 mit BARON und LindoGlobal für das mehrperiodige Modell**

Es wurde auf allen 25 Instanzen gemäß des Abschnitts 7.1 eine starre Planung für das mehrperiodige Erwartungswertmodell sowohl mit CONOPT3 (im Zuge der Untersuchungen des Abschnitts 7.2.1) als auch mit BARON und LindoGlobal durchgeführt. Bei keiner der Instanzen wurde innerhalb eines Zeitlimits von 10.000 Sekunden eine global optimale Lösung gefunden. Die Zielfunktionswerte für die bis dahin gefundene beste Lösung entsprachen denen von CONOPT3.

## **7.3 Vergleich verschiedener Planungsansätze**

Wir möchten nun die flexible Planung unter Anwendung des DDPG-Algorithmus mit der rollierenden Planung und einer zufälligen Budgetallokation vergleichen. Der Vergleich findet im Rahmen eines Meta-Algorithmus statt, den wir im Unterabschnitt 7.3.1 besprechen. Mit diesem Meta-Algorithmus erzielen wir in Abschnitt 7.3.2 die Rechenergebnisse für den Vergleich der verschiedenen Planungsansätze. Um genauer zu untersuchen, in welchen Parameterkonfigurationen welcher der Planungsansätze jeweils besser funktioniert, führen wir in Abschnitt 7.3.3 eine Regressionsanalyse durch.

### **7.3.1 Darstellung des Meta-Algorithmus**

Wir beschreiben zunächst den Zufallsalgorithmus, der als Vergleichsmöglichkeit fungiert, und stellen im Anschluss die Schritte des Meta-Algorithmus dar, der die flexible Planung mit der rollierenden Planung und der zufälligen Budgetallokation vergleicht. Danach betrachten wir die Hardware und die verwendeten Software-Pakete sowie insbesondere

die Actor-Critic-Architektur und die Hyperparameter, die die Implementierung des DDPG-Algorithmus nutzt.

### Zufällige Budgetallokation

Um eine Vergleichsmöglichkeit zu haben, wird ein Zufallsalgorithmus verwendet. Dieser wird in jeder Episode zufällig das Gesamtbudget im Sinne einer starren Planung über alle Perioden und Komponenten verteilen, sodass gilt:

$$\sum_{i=1}^n \sum_{t=1}^T x_{it} = \hat{B}.$$

Zu diesem Zweck werden die Werte  $y_{it}$  eines Stichprobenergebnisses vom Umfang  $n \cdot T$  von  $\mathcal{U}[0; 1]$ -gleichverteilten Zufallsvariablen gezogen und gemäß

$$x_{it} = \frac{y_{it}}{\sum_{j=1}^n \sum_{t'=1}^T y_{jt'}} \cdot \hat{B}$$

transformiert.

### Schritte des Meta-Algorithmus

Der übergeordnete Algorithmus, der verwendet wurde, um die Ergebnisse des folgenden Abschnitts 7.3.2 zu erzielen, ist in Abb. 7.4 dargestellt. Er besteht aus drei aufeinander aufbauenden Phasen, wobei die rollierende Planung mittels GAMS/CONOPT in Abschnitt 4.4 dieser Arbeit genauer beschrieben ist, der DDPG-Algorithmus in Alg. 6.1 auf Seite 85. Die einzelnen Phasen werden für jede der in Abschnitt 7.1 beschriebenen Instanzen nacheinander durchlaufen.

In Phase 1 werden 250 Episoden der rollierenden Planung durchgeführt. Es werden für die jeweiligen Instanzen sowohl die Realisationen des stochastischen Verschleißes als auch die Aktionen, die in jeder Periode  $t$  der einzelnen Episoden durchgeführt wurden, abgespeichert.

In Phase 2 belegen diese abgespeicherten 250 Episoden den Replay-Puffer (vgl. Abschnitt 6.3.2) mit sinnvollen Lernbeispielen für den DDPG-Algorithmus und erleichtern

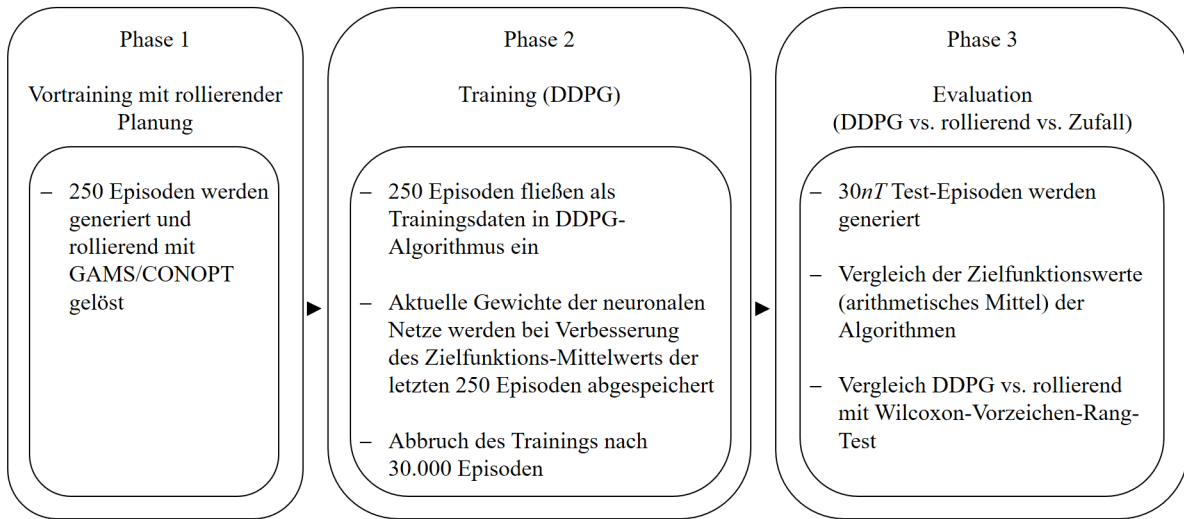


Abb. 7.4: Ablauf des Meta-Algorithmus für die vergleichende experimentelle Performance-Analyse der Planungsansätze

Quelle: Eigene Darstellung

somit den Trainingsstart.<sup>6</sup> Nach jeder Episode erhält man einen Zielfunktionswert, der aufgrund der unterschiedlichen Realisationen der Verschleiß-Zufallsvariablen jedoch stark schwankt. So wird in jeder Episode der Mittelwert der Zielfunktionswerte der letzten 250 Episoden ermittelt und, falls dieser sich verbessert hat, werden die Gewichte des zugehörigen aktuell besten Netzes – und somit eine Politik für die flexible Planung – abgespeichert. Das Training wird nach 30.000 Episoden<sup>7</sup> beendet, da Tests gezeigt haben, dass nach dieser Anzahl von Trainings keine deutliche Verbesserung mehr erzielt werden kann.

In Phase 3 findet die Evaluation statt, in der die arithmetischen Mittel der Zielfunktionswerte der Planungsansätze miteinander verglichen werden. Für die flexible Planung wird die beste bis dahin gefundene Politik genutzt, wobei das Explorationsrauschen (vgl. Abschnitt 6.3.2) dann nicht mehr verwendet wird. Für die Berechnung der arithmetischen Mittel wurden bei den fünfkomponentigen Instanzen insgesamt 750 Episoden einbezogen, bei den achtkomponentigen Instanzen 1200 Episoden. Die Anzahl der Episoden berechnet

<sup>6</sup>Ein Training ohne die Einbeziehung der vortrainierten 250 Episoden ist ebenfalls möglich.

<sup>7</sup>Bei den Parallelsystemen wurden nur 3.000 Episoden gerechnet, da das Training schon sehr schnell in den Bereich des maximal möglichen Werts 1 gekommen ist und somit Overfitting (vgl. Zhang u. a. (2018)) vermieden werden sollte.

sich jeweils aus dem Produkt

$$30 \cdot n \cdot T,$$

wobei  $n$  die Menge der Komponenten und  $T$  die Länge des Planungszeitraumes darstellt. Die Zahl 30 ist eine oft gewählte Stichprobengröße aus der Praxis, da ab dieser Größenordnung die Studentsche  $t$ -Verteilung mit der Normalverteilung hinreichend genau approximiert werden kann.

Der Vergleich von flexibler Planung mit DDPG und rollierender Planung wird zudem statistisch über den Wilcoxon-Vorzeichen-Rang-Test (vgl. Wilcoxon (1945)) abgesichert, um eine Einschätzung der Aussagekraft der Mittelwert-Vergleiche der beiden Planungsansätze zu erhalten. Dabei bedeutet ein  $p$ -Wert des Tests nahe Null, dass mit sehr hoher Wahrscheinlichkeit der eine Algorithmus dem anderen für die entsprechende Instanz überlegen ist. Ab einem  $p$ -Wert von 0,05 oder höher kann eine statistische Signifikanz angezweifelt werden.

## Hardware und Software-Pakete für die Implementierung

Die Programmierung der Algorithmen wurde mit Python in der Version 3.7.6 64-bit auf einem Notebook PC mit „Intel® Core™ i7-8650U CPU @ 1.90GHz“-Prozessor und 16 GB Arbeitsspeicher in der Entwicklungsumgebung VS-Code (Windows 10 Enterprise, 64bit) durchgeführt. Für die Lösung der mehrperiodigen Optimierungsprobleme haben wir auf den in GAMS in der Version 31.2 integrierten Solver CONOPT3 (s. Drud (2020)) mittels einer Python-Schnittstelle (vgl. GAMS Development Corporation (2021)) zugegriffen.

Des Weiteren wurden die Python-Programmbibliotheken NumPy für mathematische Berechnungen, Pandas zur Handhabung von Tabellenstrukturen, SciPy für den Wilcoxon-Vorzeichen-Rang-Test, scikit-learn für die Lasso-Regression, Matplotlib.pyplot und Seaborn für die grafische Ausgabe sowie TensorFlow / keras zur Handhabung (tiefer) neuronaler Netze eingebunden.<sup>8</sup> Eine komplette Liste der in der virtuellen Umgebung installierten Python-Pakete finden sich in Anhang B.4 auf Seite 182.

---

<sup>8</sup>Aufgrund der Verwendung von TensorFlow / keras kann allerdings eine vollständige Reproduzierbarkeit der Ergebnisse nicht sicher gestellt werden (vgl. Anhang B.3).



## Actor-Critic-Architektur und Hyperparameter

In dieser Arbeit wird gemäß Lillicrap u. a. (2015) eine Actor-Critic-Architektur verwendet. Die Hyperparameter und der Aufbau der zugehörigen neuronalen Netze sollen hier kurz erläutert werden.

Das Gedächtnis für die zu lernenden Beispiele (Replay-Puffer) hat eine Kapazität von 50.000 Einträgen („buffer size“). Es werden aus diesem Replay-Puffer bei jeder Iteration 32 Samples („batch size“) entnommen. Es wird aufgrund des verhältnismäßig kurzen Planungszeitraums ein Diskontierungsfaktor von  $\gamma = 1$  angenommen. Die Anwendung des vorgestellten Deep Reinforcement Learning-Algorithmus ist aber ohne weiteres auch auf längere Planungszeiträume erweiterbar, wo typischerweise  $\gamma < 1$  gewählt werden würde.

Es wird der Adam Optimizer (vgl. Kingma und Ba (2014)) genutzt, der zum Zeitpunkt der Erstellung der vorliegenden Dissertation als Stand der Technik gilt. Für die Start-Lernrate wurde  $lr = 5 \cdot 10^{-5}$  gewählt. Der Update-Parameter für die Ziel-Netzwerke liegt bei  $\delta = 0.001$ .

Der Aufbau des Actor-(Feed-Forward-)Netzwerks ist wie folgt:

1. Input-Schicht mit Eingabeinheiten
2. Versteckte Schicht: Vollständig (mit der nächsten Schicht) vernetzte Schicht mit 128 Neuronen und ReLU-Aktivierungsfunktion<sup>a</sup> (vgl. Nwankpa u. a. (2018))
3. Versteckte Schicht: Gauß'sches Rauschen (Standardabweichung 1,0)
4. Versteckte, vollständig (mit der nächsten Schicht) vernetzte Schicht mit 128 Neuronen und ReLU-Aktivierungsfunktion
5. Versteckte Schicht: Gauß'sches Rauschen (Standardabweichung 1,0)
6. Output-Schicht: Vollständig (mit den Ausgabeneuronen) vernetzte Schicht mit Softmax-Aktivierungsfunktion<sup>b</sup>, die einen Wertebereich von  $[0; 1]^{n+1}$  zurückliefert.

---

<sup>a</sup>ReLU ist eine nichtlineare Aktivierungsfunktion. Diese Funktion kann wie folgt dargestellt werden:  
 $f(x) = \max\{0, x\}$ .

<sup>b</sup>Die Softmax-Funktion ist eine Verallgemeinerung der logistischen Funktion auf mehrere Dimensionen:  
 $f : \mathbb{R}^{n+1} \rightarrow \mathbb{R}^{n+1}, f_i(\mathbf{x}) = \frac{\exp(x_i)}{\sum_{j=1}^{n+1} \exp(x_j)}$ .

Für den Aufbau des Critic-(Feed-Forward-)Netzwerks wurde folgende Konfiguration gewählt:

1. Input-Schicht mit Eingabeinheiten
2. Versteckte schicht: Vollständig (mit der nächsten Schicht) vernetzte Schicht mit 256 Neuronen und ReLU-Aktivierungsfunktion
3. Zusammenführung mit Ergebnis aus Actor-Netz
4. Versteckte, vollständig (mit der nächsten Schicht) vernetzte Schicht mit 128 Neuronen und ReLU-Aktivierungsfunktion
5. Output-Schicht: Vollständig (mit den Ausgabeneuronen) vernetzte Schicht mit linearer Aktivierungsfunktion

Die verrauschten Schichten finden ausschließlich zur Stabilisierung des Trainings Anwendung, in der Evaluationsphase wird das Gauß'sche Rauschen abgeschaltet.

### 7.3.2 Rechenergebnisse

Wir stellen eine Übersicht über die gesamten Ergebnisse der gerechneten Instanzen zunächst tabellarisch dar und bewerten die Planungsansätze „rollierend“ und „flexibel mit DDPG“ gegenüber der zufälligen Budgetallokation. Danach vergleichen wir die beiden Planungsansätze untereinander. Im Anschluss findet eine genauere Betrachtung derjenigen Instanzen statt, bei denen der DDPG-Algorithmus besonders gut bzw. besonders schlecht gegenüber der rollierenden Planung abgeschnitten hat.

#### Übersicht

In Tabelle 7.3 auf Seite 109 sind die Mittelwerte der Zielfunktionswerte für die verschiedenen Planungsansätze, ein Vergleich von zufälligen Investitionen mit den anderen beiden Algorithmen sowie für den Vergleich der flexiblen Planung mit der rollierenden Planung die prozentuale Differenz für die jeweiligen Instanzen aufgelistet. Daneben findet sich der

um Optimierungspotenziale  $z^{\text{DDPG}} - z^{\text{Zufall}}$  bzw.  $z^{\text{rollierend}} - z^{\text{Zufall}}$  bereinigte Vergleich der rollierenden und flexiblen Planung, der sich durch

$$\frac{z^{\text{DDPG}} - z^{\text{rollierend}}}{\frac{1}{2}(z^{\text{rollierend}} + z^{\text{DDPG}}) - z^{\text{Zufall}}}$$

ausdrücken lässt.<sup>9</sup> Wenn es in der Bezeichnung der Instanz nicht anders erwähnt ist, handelt es sich um Instanzen mit jeweils fünf Komponenten.

### Vergleich mit zufälliger Budgetallokation

Für den Zufallsalgorithmus sehen wir, dass er gegenüber der rollierenden Planung und gegenüber der flexiblen Planung in keiner der getesteten Instanzen einen besseren Zielfunktionswert erzielt hat. Im Mittel liefert der Zufallsalgorithmus über die getesteten Instanzen hinweg ca. 8,6 % schlechtere Zielfunktionswerte als die rollierende Planung und 7,7 % schlechtere als die flexible Planung mit DDPG (vgl. Tab. 7.3). Grundsätzlich ist damit der Einsatz sowohl von der rollierenden als auch von der flexiblen Planung mit DDPG bei der periodenübergreifenden Budgetallokation modularer Systeme zu empfehlen.

### Vergleich flexible Planung und rollierende Planung

Beim Vergleich von DDPG-Algorithmus und rollierender Planung fällt auf, dass der DDPG-Algorithmus im Mittel 1,0 % schlechtere Ergebnisse erzielt als die rollierende Planung, im Median hingegen nur 0,7 % schlechtere Ergebnisse. Allerdings gibt es Instanzen (z. B. Seriensystem, Nr. 1), bei denen die flexible Planung einen um 5,1 % besseren Zielfunktionswert im Mittel über 750 Episoden aufweisen kann. Man kann anhand der Tabelle 7.3 auf Seite 109 vermuten, dass die Systemstruktur Einfluss darauf hat, welche Art der Planung besser funktioniert. Der Vergleich über die strukturspezifischen Mittelwerte in Tab. 7.4 auf Seite 110 bestätigt dies: Bei den Seriensystem-Instanzen ist der Zielfunktionswert der flexiblen Planung mit DDPG im Mittel über die gerechneten

---

<sup>9</sup>Die Kennzeichnung mit einem Stern bedeutet, dass bei dieser Instanz der Wilcoxon-Vorzeichen-Rang-Test einen p-Wert  $< 0,05$  ausgewiesen hat und die rollierende Planung gegenüber der flexiblen mit DDPG im Vorteil ist, die Kennzeichnung mit einem Pluszeichen analog, dass die flexible Planung einen besseren Zielfunktionswert erreicht hat.

Instanzen um 0,4 % höher als derjenige der rollierenden Planung, beim Parallelsystem hingegen um 2,9 % niedriger. Dem wollen wir weiter nachgehen und sehen uns dazu nachfolgend die beiden Einzel-Instanzen genauer an, bei denen der DDPG-Algorithmus einen besonders guten bzw. besonders schlechten mittleren Zielfunktionswert gegenüber der rollierenden Planung ermittelt hat.

### **Einzel-Instanzen mit den größten Zielfunktionswert-Unterschieden von rollierender und flexibler Planung**

Die beiden Instanzen, die den besten bzw. schlechtesten Vergleichswert zwischen dem DDPG-Algorithmus und der rollierenden Planung aufweisen, sind die fünfkomponentige Serienschaltung Nr. 1 (s. Abb. 7.5 auf der nächsten Seite) und die fünfkomponentige Parallelschaltung Nr. 4 (s. Abb. 7.6 auf Seite 110). In den Grafiken wurde aus Übersichtlichkeitsgründen nur der Zielfunktionswert jeder zehnten Episode geplottet. Die gestrichelten Linien zeigen jedoch die jeweiligen Mittelwerte über alle 750 Episoden.

Es fällt auf den ersten Blick auf, dass die Schwankungen um die Mittelwerte in Abb. 7.5 für alle Planungsansätze recht groß ausfallen, was darauf hindeuten könnte, dass es schwieriger ist, bei Instanzen mit hohen Verschleißschwankungen auch hohe Zielfunktionswerte zu erreichen. Abb. 7.6 lässt eine alternative Vermutung zu: Hier erreicht die rollierende Planung Zielfunktionswerte, die durchgängig im Bereich des maximal möglichen Werts 1 liegen. Möglicherweise ist also der Abstand des Zielfunktionswerts zum maximal möglichen Wert ebenfalls ein Kriterium dafür, ob der DDPG-Algorithmus gut oder schlecht im Vergleich zur rollierenden Planung funktioniert. Der Abstand des Zielfunktionswerts zum maximal möglichen Wert könnte somit als eine Kennzahl dafür angesehen werden, wie schwierig das Planungsproblem ist. Dies wollen wir im folgenden Abschnitt im Rahmen einer Regressionsanalyse genauer untersuchen.

Alle weiteren Grafiken zu den einzelnen Instanzen finden sich in Anhang B.5 ab Seite 183.

### **7.3.3 Regressionsanalyse**

Im Folgenden beschreiben wir mit der Lasso-Regression diejenige Methode, die verwendet wurde, um eine Einflussanalyse der verschiedenen Instanzparameter durchzuführen. Wir tun dies, um herauszufinden, woran es liegt, dass der DDPG-Algorithmus (im Mittel)

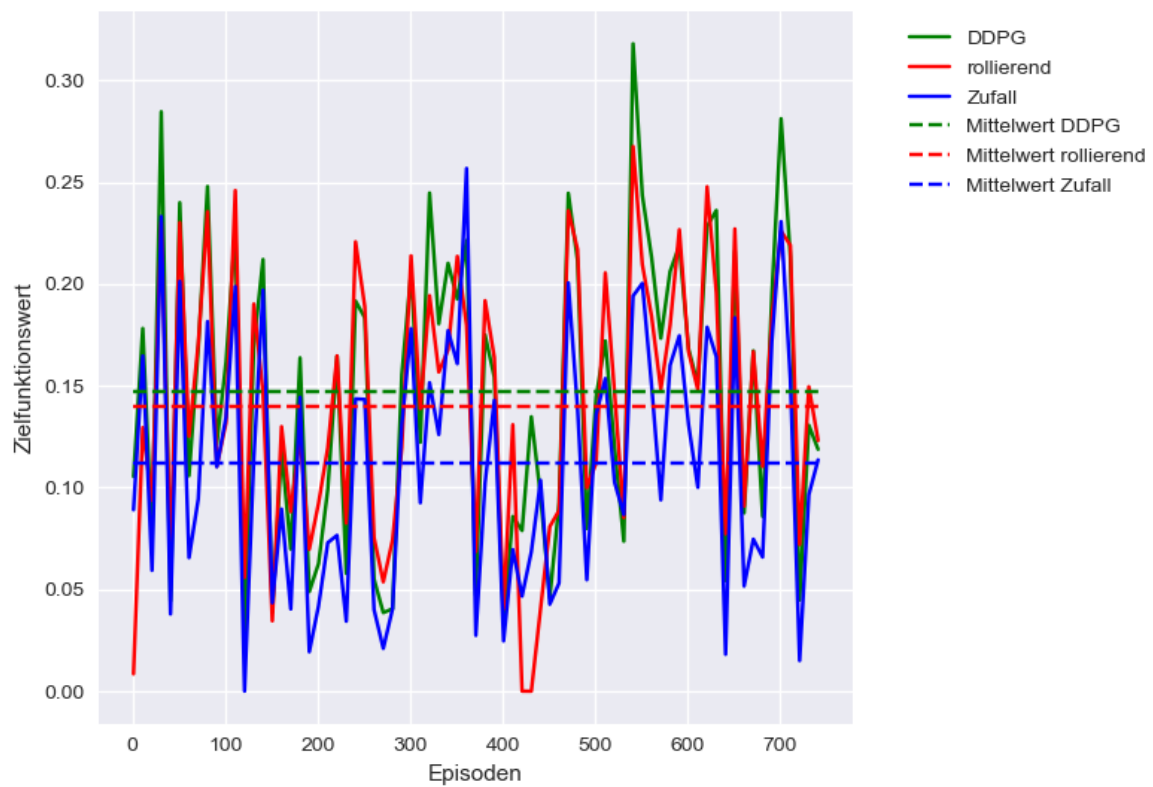


Abb. 7.5: Instanz Nr. 1 Seriensystem

Quelle: Eigene Darstellung

Instanz	$z^{\text{DDPG}}$	$z^{\text{rollierend}}$	$z^{\text{Zufall}}$	$\frac{z^{\text{Zufall}} - z^{\text{DDPG}}}{z^{\text{DDPG}}}$	$\frac{z^{\text{Zufall}} - z^{\text{rollierend}}}{z^{\text{rollierend}}}$	$\frac{z^{\text{DDPG}} - z^{\text{rollierend}}}{z^{\text{rollierend}}}$	bereinigter Vgl.
Brückenschaltung, Nr. 1*	0,799	0,826	0,766	-4,1%	-7,2%	-3,2%	-57,1%
Brückenschaltung, Nr. 2*	0,908	0,910	0,873	-3,8%	-4,0%	-0,2%	-5,3%
Brückenschaltung, Nr. 3*	0,743	0,748	0,711	-4,3%	-4,9%	-0,7%	-14,2%
Brückenschaltung, Nr. 4	0,809	0,811	0,742	-8,3%	-8,4%	-0,1%	-1,7%
Brückenschaltung, Nr. 5 <sup>+</sup>	0,848	0,847	0,801	-5,5%	-5,3%	+0,2%	+3,7%
Seriensystem, Nr. 1 <sup>+</sup>	0,147	0,140	0,112	-23,6%	-19,7%	+5,1%	+23,1%
Seriensystem, Nr. 2	0,176	0,173	0,138	-21,7%	-20,5%	+1,6%	+7,4%
Seriensystem, Nr. 3*	0,273	0,279	0,237	-13,0%	-14,9%	-2,1%	-15,4%
Seriensystem, Nr. 4	0,199	0,199	0,164	-17,6%	-17,5%	+0,2%	+1,0%
Seriensystem, Nr. 5*	0,148	0,153	0,114	-23,0%	-25,1%	-2,7%	-11,4%
Parallelsystem, Nr. 1*	0,975	1,000	0,976	0,0%	-2,4%	-2,5%	-206,0%
Parallelsystem, Nr. 2	0,984	1,000	0,969	-1,6%	-3,1%	-1,6%	-66,0%
Parallelsystem, Nr. 3*	0,976	0,999	0,908	-7,0%	-9,1%	-2,3%	-29,1%
Parallelsystem, Nr. 4*	0,937	0,998	0,867	-7,5%	-13,1%	-6,1%	-60,2%
Parallelsystem, Nr. 5*	0,981	1,000	0,939	-4,3%	-6,1%	-1,9%	-36,0%
Serien-Parallel-System, Nr. 1*	0,604	0,611	0,559	-7,5%	-8,6%	-1,2%	-14,6%
Serien-Parallel-System, Nr. 2*	0,643	0,644	0,601	-6,5%	-6,6%	-0,0%	-0,7%
Serien-Parallel-System, Nr. 3*	0,786	0,790	0,754	-4,0%	-4,6%	-0,6%	-13,9%
Serien-Parallel-System, Nr. 4*	0,751	0,756	0,706	-6,0%	-6,7%	-0,7%	-10,7%
Serien-Parallel-System, Nr. 5 <sup>+</sup>	0,737	0,733	0,663	-10,0%	-9,5%	+0,6%	+5,6%
Brückensch. (8 Komp.), Nr. 1*	0,953	0,958	0,938	-1,6%	-2,1%	-0,6%	-30,0%
Brückensch. (8 Komp.), Nr. 2*	0,907	0,920	0,875	-3,5%	-4,9%	-1,5%	-34,9%
Brückensch. (8 Komp.), Nr. 3*	0,956	0,965	0,948	-0,8%	-1,8%	-1,0%	-76,9%
Brückensch. (8 Komp.), Nr. 4*	0,889	0,916	0,853	-4,1%	-6,9%	-2,9%	-53,8%
Brückensch. (8 Komp.), Nr. 5*	0,951	0,957	0,932	-2,0%	-2,6%	-0,6%	-27,0%
Mittelwert über alle Instanzen		–		-7,7%	-8,6%	-1,0%	-29,0%
Median über alle Instanzen		–		-7,8%	-8,7%	-0,7%	-14,6%

Tab. 7.3: Mittelwerte der Zielfunktionswerte für die Instanzen im Überblick

Quelle: Eigene Darstellung

Systemstruktur	Brücken- schaltung	Seriensys- tem	Parallelsys- tem	Serien-Pa- rallel-Sys- tem	Brückensch. (8 Komp.)
Mittelwert von $\frac{z_{DDPG} - z_{rollierend}}{z_{rollierend}}$ über die Instanzen der Systemstruktur	-0,8%	+0,4%	-2,9%	-0,4%	-1,3%

Tab. 7.4: Mittelwerte des relativen Zielfunktionswerte-Vergleichs von der rollierenden Planung und der flexiblen Planung, gruppiert nach Systemstrukturen  
Quelle: Eigene Darstellung

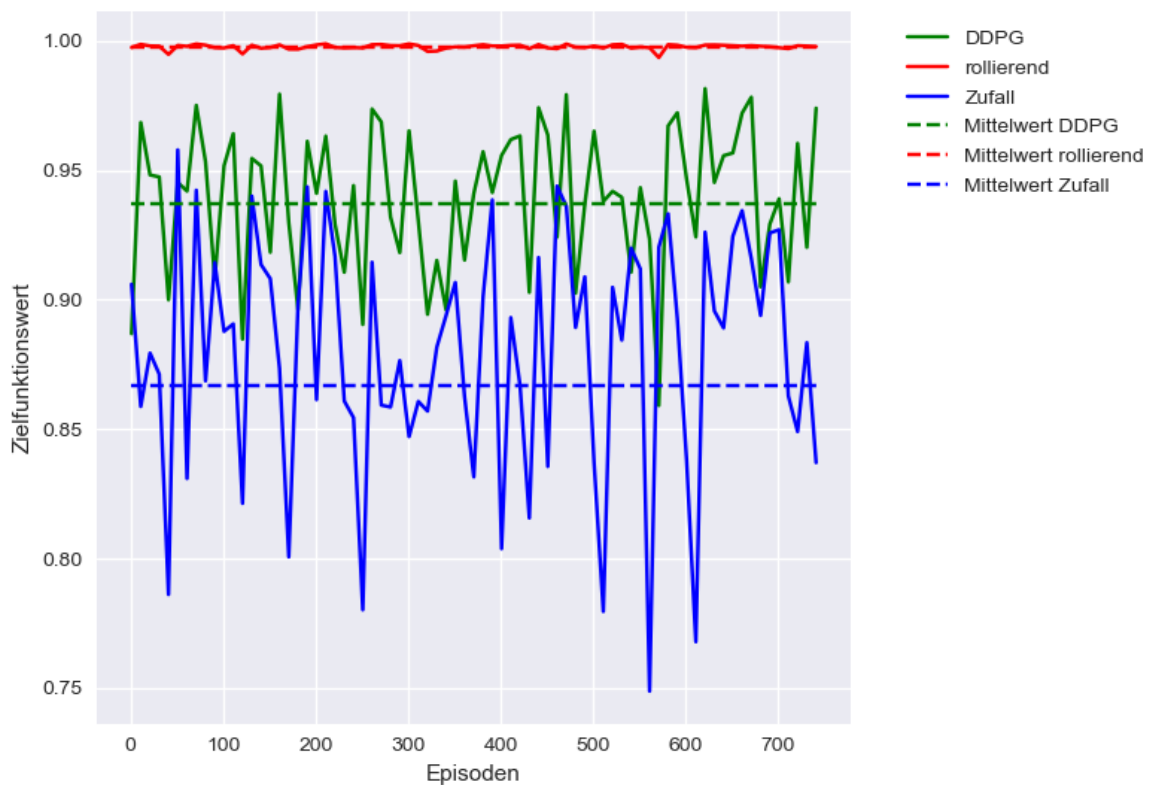


Abb. 7.6: Instanz Nr. 4 Parallelsystem  
Quelle: Eigene Darstellung

besser oder schlechter als die rollierende Planung funktioniert. Der zugehörige Python-Programmcode findet sich in Anhang B.2.3 auf den Seiten 176 ff.

## Lasso-Regression

Die Lasso-Regression als eine Form von multipler linearer Regression hat gegenüber anderen Regressionsmethoden den Vorteil, dass möglichst wenige Einflussgrößen, dafür aber welche mit höherer Aussagekraft ermittelt werden. Die Linearitätsannahme wurde getroffen, da keine Hinweise dafür vorliegen, dass eine andere Regressionsform bessere Erkenntnisse liefern könnte. Lasso steht für „Least absolut shrinkage and selection operator“ (vgl. Santosa und Symes (1986); Tibshirani (1996)). Die Lasso-Regression (bzw. „L1-Regularisierung“) ist gegenüber einer weiteren verbreiteten Regressionsmethode, der Ridge-Regression (auch bekannt unter Tikhonov-Regularisierung, vgl. Golub u. a. (1999)), mathematisch aufwändiger zu optimieren. Es gibt keine explizite analytische Darstellung des Lasso-Schätzers, und das dahinterliegende Minimierungsproblem wird über Algorithmen wie beispielsweise den „Least Angle Regression Algorithmus“ („LARS“; vgl. Tibshirani u. a. (2004); Hastie u. a. (2007)) gelöst. Lasso- und Ridge-Regression sind sich ähnlich. Beide versuchen, einen Term zu minimieren, der sowohl die Residuenquadratsumme enthält als auch einen weiteren Term, der mit der Größe der Koeffizienten wächst. Mit der Lasso-Methode werden die Koeffizienten so gewählt, dass der Ausdruck

$$\sum_{i=1}^{n_1} \left( y_i - \beta_0 - \sum_{j=1}^{n_2} \beta_j x_{ij} \right)^2 + \alpha \cdot \sum_{j=1}^{n_2} |\beta_j|$$

minimiert wird. Hierbei ist der quadratische Term die Residuenquadratsumme,  $y_i$  ist die abhängige Variable für eine der  $n_1$  Trainingsdatensätze und  $n_2$  die Anzahl der unabhängigen Variablen. Somit stellen die  $\beta_j$  die zu optimierenden Koeffizienten dar. Der Parameter  $\alpha$  kann verschiedene Werte zwischen 0 und 1 annehmen. Wird  $\alpha = 0$  gewählt, so erfolgt die Regression wie bei der einfachen multivariablen Regression. Wird  $\alpha = 1$  gewählt, so werden alle Koeffizienten zu Null.



## Eingangsdaten

Als Trainingsdaten flossen die Mittelwerte über die Instanzparameter der Komponenten<sup>10</sup> als ein Maß für die Lage und die Standardabweichungen als ein Maß für die Streuung sowie die Anzahl der Komponenten ein. Die Systemstruktur ging des Weiteren in einer sog. „One-Hot-Kodierung“ ein, da die unabhängige Variable „Systemstruktur“ kategorisch ist, also keine Rangordnung unter den einzelnen Strukturen besteht. Um zu untersuchen, ob der Abstand zum maximal möglichen Wert 1 einen Einfluss hat, fügen wir eine weitere unabhängige Variable hinzu: die Differenz  $1 - z^{\ddot{u}}$ , wobei wir  $z^{\ddot{u}}$  der Tab. 7.1 auf Seite 97 entnehmen. Dies sind also die Abstände von den maximal möglichen Werten der von uns betrachteten Strukturbewertungsfunktionen (und damit auch der Zielfunktionen) vom Zielfunktionswert des Erwartungswertmodells der starren Planung. Als abhängige Variable wurde die prozentuale Abweichung des DDPG-Algorithmus gegenüber der rollierenden Planung gewählt.

Um Vergleichbarkeit zwischen den Eingangsdaten herzustellen, findet vor der eigentlichen Lasso-Regression eine Normalisierung dieser Daten statt, indem der entsprechende Mittelwert subtrahiert und durch die  $\ell^2$ -Norm<sup>11</sup> geteilt wird. Eine Übersicht der Trainingsdaten findet sich in Tab. B.6 auf Seite 206 im Anhang.

## Ergebnis der Regression

Als Lösungsmethode wurde mit Hilfe des Python-Pakets „scikit-learn“ der LARS-Algorithmus verwendet, und zwar mit einer Kreuzvalidierung, die die Trainingsdaten in fünf gleiche Teile einteilt und somit in fünf Durchgängen auf jeweils 80 % der Daten trainiert und auf den verbleibenden 20 % der Daten auf Generalisierbarkeit testet. Dadurch wird der Regularisierungsparameter bestmöglich automatisiert gewählt.

In Tabelle 7.5 sind die Koeffizienten der unabhängigen Variablen, die einen Wert ungleich Null aufweisen, aufgelistet<sup>12</sup> sowie das Bestimmtheitsmaß  $R^2$ . Es wird die Vermutung bestätigt, dass der Abstand vom maximal möglichen Wert oder die Systemstruktur einen

---

<sup>10</sup>Dies ist z. B. für die Skalierungsparameter der Investitionsfunktion der Ausdruck  $\frac{1}{n} \sum_{i=1}^n \hat{a}_i^j$  für eine bestimmte Instanz  $j$  mit  $n$  Komponenten.

<sup>11</sup>Die  $\ell^2$ -Norm ist definiert als  $\|(x_i)_{i=1,\dots,n}\|_{\ell^2} := (\sum_{i=1}^n x_i^2)^{\frac{1}{2}}$ .

<sup>12</sup>Der Regularisierungsparameter hat auf vier Nachkommastellen gerundet den Wert  $\alpha = 0,0010$ , der Versatz  $\beta_0 = -0,0106$ . Für die Interpretation sind diese Werte allerdings irrelevant.

Einfluss darauf haben können, ob der DDPG-Algorithmus besonders gut oder besonders schlecht gegenüber der rollierenden Planung abschneidet. Ein negativer Wert von  $\beta_{\text{par}}$  deutet darauf hin, dass die rollierende Planung der flexiblen Planung mit DDPG bei der Systemstruktur „Parallelsystem“ überlegen ist. Ein positiver Wert von  $\beta_{1-z^{\ddot{u}}}$  bedeutet, dass ein größerer Abstand vom maximal möglichen Wert eher für den Einsatz der flexiblen Planung mit DDPG spricht.

$\beta_{1-z^{\ddot{u}}}$	$\beta_{\text{par}}$	$R^2$
0,0102	-0,0084	0,2249

Tab. 7.5: Relevante Koeffizienten der Regressionsanalyse mit LARS, gerundet auf vier Nachkommastellen

Quelle: Eigene Darstellung

Diese Aussagen haben allerdings eine begrenzte Aussagekraft, da das Bestimmtheitsmaß Werte zwischen 0 und 1 annehmen kann und prinzipiell dabei niedrigere Werte (in diesem Fall 0,2249) für eine geringere Vorhersagekraft der abhängigen Variable stehen.

### Beispielhafte Instanzanpassung

Um die vorangegangenen Vermutungen zu validieren, wurde die Instanz Nr. 1 der fünfkomponentigen Brückenschaltung (s. Abb. 7.7 auf der nächsten Seite) bewusst verändert. Laut Ergebnis-Tabelle 7.3 auf Seite 109 ist bei dieser Instanz der Zielfunktionswert der flexiblen Planung mit DDPG im Mittel 3,2 % schlechter als bei der rollierenden Planung.

Es wird nun eine weitere Instanz aus dieser heraus durch Parameteranpassung konstruiert, um bewusst ein verbessertes Verhalten der flexiblen Planung mit DDPG herbeizuführen. Dazu wurde versucht, den Zielfunktionswert  $z^{\ddot{u}}$ , der nach Tab. 7.1 auf Seite 97 bei 0,8712 liegt, bewusst zu verkleinern, um den Abstand zum maximal möglichen Wert 1 zu vergrößern. Zu diesem Zweck wurde der Verschleiß dieser Instanz für die einzelnen Komponenten erhöht, wie Tab. B.7 auf Seite 207 des Anhangs zu entnehmen ist,<sup>13</sup> und zwar durch im Mittel leichte Erhöhung der Instanzparameter  $\hat{b}_i$  und starke Erhöhung der

---

<sup>13</sup>Zur Erinnerung: Der Erwartungswert des kumulierten Verschleißes bis zur Periode  $t$  wird angenähert über  $\frac{c \cdot t^b}{u}$ , die Varianz über  $\frac{c \cdot t^b}{u^2}$  (vgl. Abschnitt 4.2).

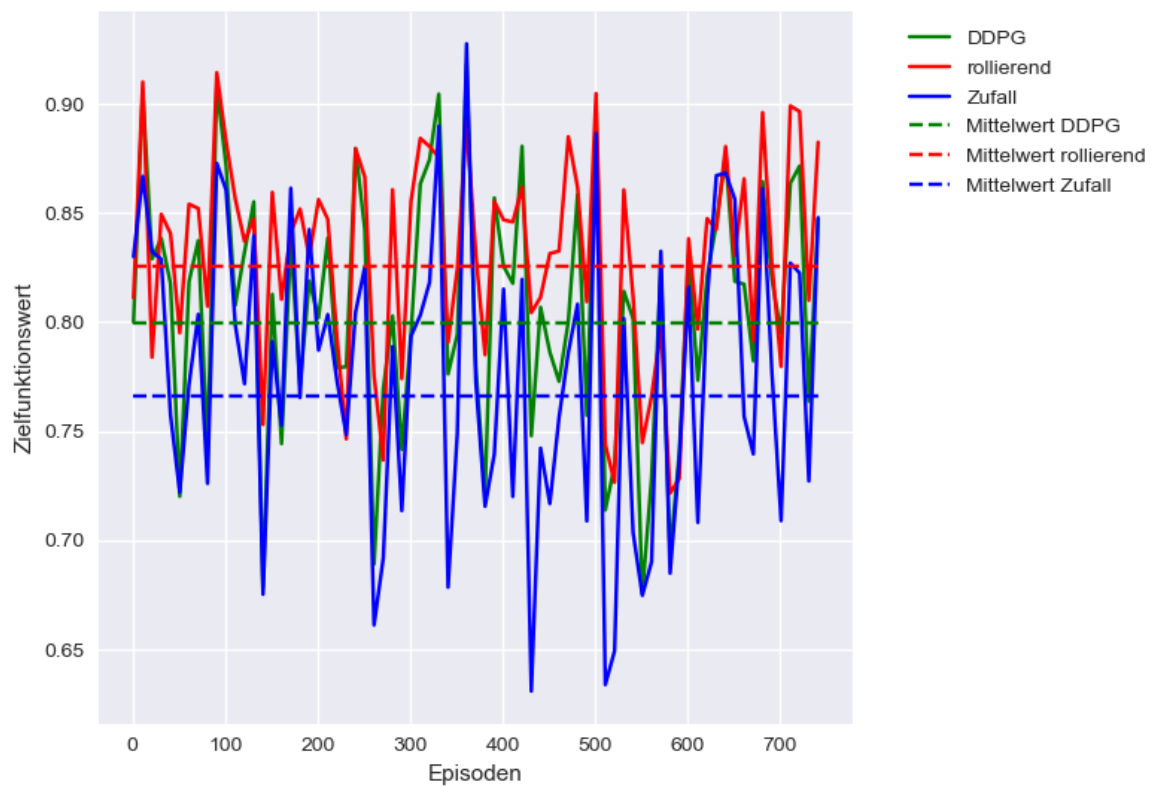


Abb. 7.7: Instanz Nr. 1 Brückenschaltung mit fünf Komponenten

Quelle: Eigene Darstellung

Instanzparameter  $\hat{c}_i$  der Komponenten  $i$ . Dadurch sinkt  $z^{\hat{u}}$  für die konstruierte Instanz von 0,8712 auf 0,3538, gleichzeitig erhöht sich die Varianz.<sup>14</sup>

Das Ergebnis der Anpassung ist grafisch in Abb. 7.8 dargestellt. Man kann erkennen, dass die konstruierte Instanz nun wie vermutet im Mittel einen höheren Zielfunktionswert für die flexible Planung mit DDPG liefert als für die rollierende Planung.

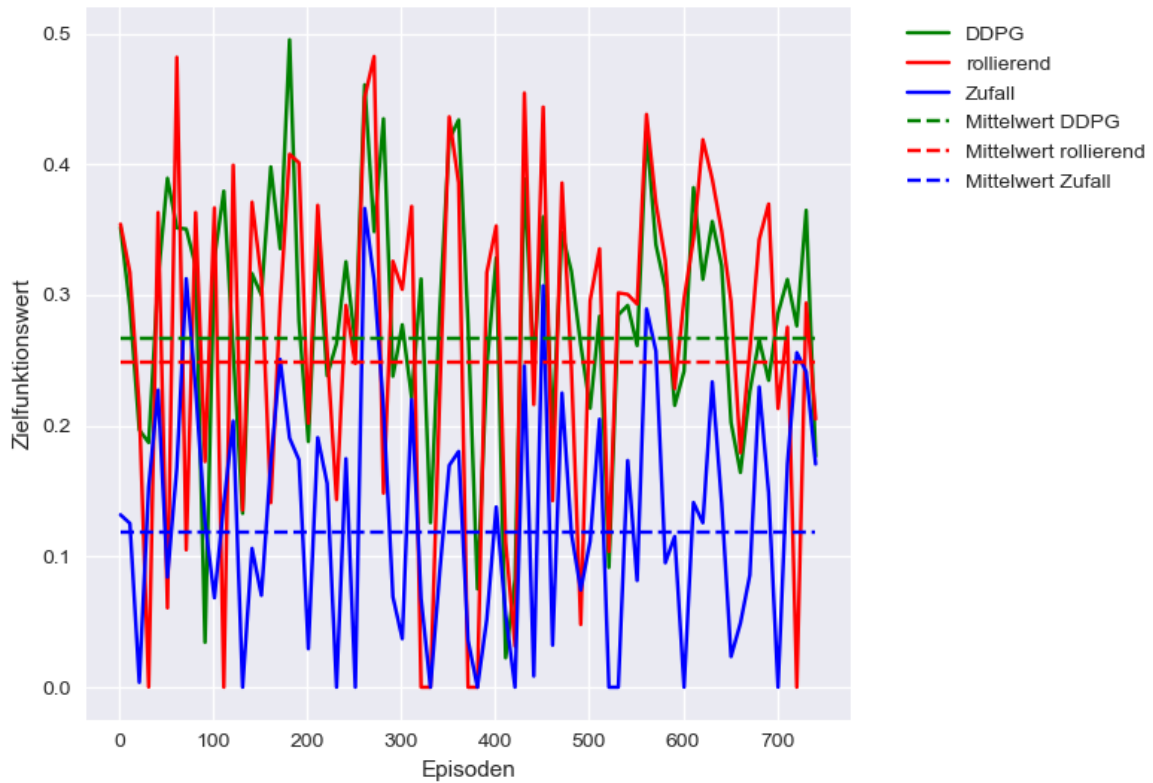


Abb. 7.8: Konstruierte Instanz Nr. 1 der Brückenschaltung mit fünf Komponenten  
Quelle: Eigene Darstellung

Der visuelle Eindruck wird durch die konkreten Zielfunktionswerte in Tab. 7.6 auf der nächsten Seite untermauert. Die konstruierte Instanz hat damit unter den gerechneten Instanzen den vergleichsweise besten Zielfunktionswert der flexiblen Planung mit DDPG gegenüber der rollierenden Planung. Diese Information kann bei der Wahl des richtigen Planungsverfahrens helfen: So kann über eine starre Planung sehr schnell abgeschätzt

<sup>14</sup>Hier wurde die Approximation der Maximumfunktion (vgl. Drud (2021b)) für die Zustandsübergangsfunktionen  $g_i$  verwendet, da ansonsten prozentuale Abnutzungsvorräte unter 0 gesunken wären.

werden, ob eine sehr gute Systemleistung mit dem verfügbaren Budget erreicht werden kann. Sollte dies möglich sein, könnte die rollierende Planung zum Einsatz kommen, andernfalls die flexible Planungsmethode mit DDPG.

Instanz	$z^{\text{DDPG}}$	$z^{\text{rollierend}}$	$\frac{z^{\text{DDPG}} - z^{\text{rollierend}}}{z^{\text{rollierend}}}$
Brückenschaltung, Nr. 1 - vorher	0,80	0,83	-3,2%
Brückenschaltung, Nr. 1 - konstruiert	0,27	0,25	+7,0%

Tab. 7.6: Mittelwerte der Zielfunktionswerte für die konstruierten Instanzen  
Quelle: Eigene Darstellung

### 7.3.4 Kritische Bewertung der Ergebnisse

Die Ergebnisse der Abschnitte 7.3.2 und 7.3.3 konnten den Vorteil, den eine flexible Planung in der Theorie gegenüber der rollierenden Planung hat, nur bedingt zeigen. Im Konzept der flexiblen Planung wird bei der Festlegung der Politik für die erste Periode bereits antizipiert, dass in den Folgeperioden Entscheidungen in Abhängigkeit der dann realisierten Verschleiß der Komponenten getroffen werden. Bei der rollierenden Planung findet diese Antizipation in der ersten Periode nur dahingehend statt, dass die Erwartungswerte für den Verschleiß bekannt sind und auf dieser Grundlage geplant wird.

Daraus folgt, dass in der Theorie eine flexible Planung schon ab der ersten Periode Schwankungen im Verschleiß für alle Planungsperioden berücksichtigen kann und somit für alle Instanzen ein höherer Zielfunktionswert im Mittel beim Vergleich der beiden Planungsansätze erwartet werden könnte. Dies geht jedoch weder aus den Daten der Tab. 7.3 auf Seite 109 noch (direkt) aus den Ergebnissen der Regressionsanalyse (vgl. Tab. 7.5 auf Seite 113) hervor. Indirekt kann der Abstand vom maximal möglichen Zielfunktionswert als Kennzahl für die Schwierigkeit des Planungsproblems die These der besseren Antizipation der flexiblen Planung stützen, da diese Kennzahl durch den zugehörigen positiven Koeffizienten in der Regressionsanalyse als unabhängige Variable identifiziert wurde. Bestätigt wird die Theorie weiterhin von den Ergebnissen der konstruierten Instanz (vgl. Abb. 7.8 auf Seite 115), bei der die erwartete Varianz des Verschleißes bewusst durch die Anpassung der Instanzparameter erhöht wurde.

Mögliche Gründe dafür, dass die Ergebnisse nur teilweise den Erwartungen entsprechen, könnten darin liegen, dass 1) die Verschleiß bei den Instanzen (bis auf die konstruierte Instanz) bewusst niedrig gewählte Erwartungswerte hatten und 2) das Training der neuronalen Netze zu lang oder zu kurz war oder 3) die Wahl der Hyperparameter oder der Netz-Architektur des DDPG-Algorithmus weiter verbessert werden müssten.

Der erwartete Verschleiß wurde bei den Instanzen so niedrig gewählt, um zumindest für die starre Planung zu gewährleisten, dass die prozentualen Abnutzungsvorräte stets streng größer als Null sind (vgl. Voraussetzungen der Optimalität in Abschnitt 5.4.1). Als demgegenüber diese Voraussetzung nicht getroffen und bewusst ein höherer Erwartungswert angenommen wurde (Konstruktion der Instanz, vgl. Tab. B.7 auf Seite 207), war der Zielfunktionswert der flexiblen Planung mit DDPG im Mittel besser.

Wenn das Training des neuronalen Netzes mit 30.000 Episoden zu kurz gewählt wurde,

kann das Verfahren die bis dahin gefundene Politik noch verbessern. Wenn das Training zu lang gewählt wurde, hat es möglicherweise an Generalisierbarkeit eingebüßt („Overfitting“). Weder die eine noch die andere Möglichkeit ging aus den Log-Daten des Trainings hervor.<sup>15</sup>

Die dritte Möglichkeit, dass die Wahl der Hyperparameter oder der Netz-Architektur noch verbessert werden könnte, ist ein reales Problem von maschinellen Lernverfahren wie Deep Reinforcement Learning, dem viele Praktiker gegenüberstehen. Nach mehreren Tests für diverse Konfigurationen der Hyperparameter (z. B. anfängliche Lernrate) und Netz-Architektur (z. B. die Anzahl der Knoten oder die Wahl der Aktivierungsfunktionen) wurde die in Abschnitt 7.3.1 beschriebene Konfiguration als beste bezüglich Laufzeit bzw. Trainingsverhalten gewählt.

---

<sup>15</sup>Mit Ausnahme der Parallelsysteme: Hier wurde die Anzahl der Episoden auf 3.000 angepasst.

# 8 Einordnung der Ergebnisse und offene Fragen

Wir wollen nun die Ergebnisse in Abschnitt 8.1 im Forschungsfeld verorten und offene Fragen in den weiteren Abschnitten dieses Kapitels besprechen. In Abschnitt 8.2 klären wir, in welchen Fällen die verwendeten Methoden adäquat sind und wo ihre Grenzen liegen. Darauf aufbauend gibt Abschnitt 8.3 Empfehlungen, wie die Ergebnisse dieser Arbeit Anwendung in der Praxis finden können. Der Forschungsbedarf wird zuletzt in Abschnitt 8.4 erörtert.

## 8.1 Einordnung der Ergebnisse

In Abschnitt 3.2.4 wurde die Forschungslücke beschrieben. Es wurde in der Literatur keine andere Arbeit identifiziert, die Budgetallokation für ein *modulares* System behandelt, dessen Komponenten einem Verschleiß unterliegen, für den eine *nichtstationäre* Verteilungsannahme zugelassen ist, wenn ein Gesamtbudget *periodenübergreifend* für einen mehrperiodigen Planungszeitraum zur Verfügung steht. Die in dieser Arbeit vorgestellten Planungsansätze der rollierenden und flexiblen Planung mit DDPG konnten Wege aufzeigen, wie eine geeignete Budgetallokation ermittelt werden kann. Die rollierende Planung begegnet den zeitabhängigen Verschleißprozessen mit Neuplanungen in jeder Periode. Die flexible Planung simuliert hingegen tausendfach die angenommenen Verschleißprozesse und lernt aus den gewonnenen Erkenntnissen eine passende Politik.

Ein weiteres Ergebnis dieser Arbeit betrifft das Potenzial periodenübergreifender Budgetallokation. Wie in Kapitel 3 gezeigt, finden sich in der Literatur viele Arbeiten, bei denen eine periodenbezogene Budgetrestriktion vorliegt. Hier hat die Arbeit anhand einer experi-



mentellen Analyse in Form einer starren Planung gezeigt, dass eine periodenübergreifende Budgetallokation im Mittel mehr als 5% besser ist als eine periodenbezogene.

## 8.2 Methodenkritik

Wir werfen in diesem Abschnitt einen kritischen Blick auf die verwendeten Methoden der Arbeit. Die verwendeten Methoden des Kapitels 4 sind mathematische Standard-Methoden wie z. B. Widerspruchsbeweise oder die Untersuchung der Karush-Kuhn-Tucker-Bedingungen. Daher beziehen sich die weiteren Ausführungen auf die verwendeten Planungsansätze. Konkret betrachten wir in Unterabschnitt 8.2.1 die rollierende Planung und in Unterabschnitt 8.2.2 die flexible Planung. Wir bewerten dabei die unterschiedlichen Planungsansätze jeweils hinsichtlich

- Ergebnisgüte,
- Einfluss der Datenqualität,
- Rechendauer der Algorithmen,
- Erweiterbarkeit des zugrundeliegenden mathematischen Modells,
- Grenzen der Methoden und
- Alternativen zu den Methoden.

### 8.2.1 Rollierende Planung

Die folgenden Ausführungen beziehen sich auf das Erwartungswertmodell aus Abschnitt 4.4 und die Umsetzung als rollierende Planung mit dem Optimierungssolver CONOPT3.

*Ergebnisgüte* Bei als deterministisch angenommenem Verschleiß liefert die rollierende Planung für Seriensysteme stets die optimale Lösung bei perfekter Datenqualität. Generell muss festgestellt werden, dass bei stochastischen Einflussgrößen die rollierende Planung nur eine Heuristik ist und die Güte der Ergebnisse deutlich sinken kann. Zudem kann nicht ausgeschlossen werden, dass bei Strukturbewertungsfunktionen, die nicht quasikonkav sind, die Verwendung von CONOPT3

nur zu lokalen Optima führt. Grundsätzlich ist allerdings die rollierende Planung eine Heuristik und nach den Erkenntnissen dieser Arbeit bezüglich des Zielfunktionswerts deutlich besser als eine rein zufällige Investitionen.

*Einfluss der Datenqualität* Bei Auftreten von fehlerhaften Daten wie etwa Ausreißern (durch Eingabefehler o. Ä.) kann das Optimierungsmodell keine zuverlässig gute Lösung garantieren. Dadurch, dass immer nur eine Periode bei der Planung umgesetzt wird, ist allerdings der Einfluss von Ausreißern geringer als bei einer starren Planung.

*Rechendauer* Da es sich um eine taktische bzw. aggregierte Planung handelt, ist die Rechendauer für die behandelten Instanzen mit wenigen Sekunden auf einem Standard-PC vernachlässigbar.

*Erweiterbarkeit* Das Hinzufügen weiterer Komponenten und Perioden ist aufgrund der geringen Rechendauer ohne Weiteres möglich. Auch eine Übertragung auf binäre Systeme ist möglich, wenn der Zustand einer Komponente als Zuverlässigkeit aufgefasst wird und sich die angenommene Verschleißverteilung damit vereinbaren lässt.

*Grenzen* Für andere Zielfunktionen als die in dieser Arbeit behandelte Maximierung der minimalen Perioden-Systemleistung müssten weitergehende Analysen angestellt werden, ob die Verwendung der rollierenden Planung auch dann noch gute Ergebnisse liefern kann. Wenn, wie es in der Literatur in einigen Arbeiten der Fall ist (s. Kapitel 3), ein unendlicher Planungszeitraum vorausgesetzt wird, kann das Erwartungswertmodell in der aktuellen Form nicht verwendet werden. Auch ein vom Zustand abhängiger Verschleiß oder andere nicht betrachtete Formen von Abhängigkeiten (vgl. Abschnitt 2.3) können i. d. R. nicht mit dem vorliegenden Modell abgebildet werden.

*Alternativen* Für das Erwartungswertmodell des Abschnitts 4.4 gibt es weitere geeignete Optimierungssolver. Für einen Überblick über die in GAMS zur Verfügung stehenden Alternativen sei an dieser Stelle auf Andrei (2017a) verwiesen. Exakte Solver wie LindoGlobal garantieren zwar Optimalität auch für nicht-quasikonkave Strukturbewertungsfunktionen, tun dies allerdings i. d. R. auf Kosten einer höheren Rechendauer. Neben weiteren Optimierungssolvern besteht eine Alternative darin, bei der Nutzung von GAMS szenario-basierte Auswertun-

gen (z. B. Worst- oder Best-Case-Szenarien) oder Sampling-Approximationen zu betrachten. Der flexible Planungsansatz mit DDPG kann (wie in dieser Arbeit gezeigt wurde) ebenfalls eine gute Alternative sein. Eine Erweiterung der Implementierung von Algorithmus 5.1 auf das mehrperiodige Problem kann ebenfalls als Alternative in Erwägung gezogen werden, möglicherweise auf Grundlage der zugehörigen mehrperiodigen KKT-Bedingungen (s. Anhang A.3.4 auf Seite 156) oder Satz 3 auf Seite 51.

### 8.2.2 Flexible Planung mit DDPG

In diesem Unterabschnitt bewerten wir die flexible Planung mit DDPG ebenfalls hinsichtlich der sechs o. g. Kriterien.

*Ergebnisgüte* Wie Abschnitt 7.3 zeigt, kann es sinnvoll sein, anstelle der rollierenden Planung den flexiblen Planungsansatz zu verwenden. Insgesamt ist die Ergebnisgüte in den behandelten Instanzen jedoch im Mittel etwas schlechter als beim rollierenden Planungsansatz, ist aber immer noch einer rein zufälligen Planung vorzuziehen.

*Einfluss der Datenqualität* Dadurch, dass innerhalb des neuronalen Netzes eine verrauschte Schicht für das Training hinzugefügt wurde, ist der Algorithmus darauf ausgelegt, dass die Datenqualität nicht perfekt ist. Hier liegt einer der großen Vorteile des DDPG-Algorithmus.

*Rechendauer* Die Trainingsdauer für den Algorithmus ist auf einem Standard-PC sehr lang, wenn auf einer CPU gerechnet wird. Für die Testinstanzen wurden teilweise Dauern von mehreren Stunden gemessen. Da allerdings das Training nur einmal im Voraus offline durchgeführt wird und nicht jede Periode trainiert werden muss, solange sich die Verteilungsannahmen des Verschleißes nicht ändern, kann im Rahmen einer taktischen Planung diese Dauer noch als akzeptabel gelten.

*Erweiterbarkeit* Der DDPG-Algorithmus, der im Rahmen der flexiblen Planung verwendet wurde, lässt sich gut erweitern. Für größere Instanzen oder weitere Perioden kann durch minimale Anpassung des Programm-Codes auf (mehreren) Graphics Processing Units (GPUs) gerechnet werden, um ein Vielfaches an

Trainingsgeschwindigkeit zu erreichen. Sollte die entsprechende Hardware nicht verfügbar sein, können Ressourcen z. B. bei Diensten wie Amazon Web Services (AWS) gemietet werden. Wenn eine alternative Zielfunktion im Sinne eines stochastischen dynamischen Optimierungsproblems formuliert werden kann, wie z. B. die Maximierung der durchschnittlichen Perioden-Systemleistungen, ist eine Anpassung ebenfalls leicht möglich. Auch die Berücksichtigung von weiteren Abhängigkeiten im Sinne des Abschnitts 2.3 sind häufig möglich.

*Grenzen* Die Grenzen des Algorithmus liegen vor allem in zeitlichen Restriktionen für die Implementierung, die deutlich aufwändiger ist als die Nutzung eines Standard-Optimierungssolvers. Auch kann für sehr große Instanzen die Trainingsdauer auf einem handelsüblichen Rechner eine Anwendung durch lange Rechenzeiten oder zu hohen Speicherverbrauch verhindern.<sup>1</sup>

*Alternativen* Nach den vorliegenden Ergebnissen dieser Arbeit ist die rollierende Planung eine sehr gute und leicht zu implementierende Alternative zur flexiblen Planung. Zugleich entwickelt sich das Deep Reinforcement Learning rasant weiter, und es werden in kurzen Abständen stetig neue Methoden veröffentlicht, die möglicherweise für das vorliegende Problem ebenfalls eine sehr gute Lösung finden. Dies soll im Abschnitt 8.4 weiter besprochen werden.

## 8.3 Implikationen und Empfehlungen

In diesem Abschnitt stellen wir die Bedeutung der Erkenntnisse dieser Arbeit für betroffene potenzielle Nutzer dar und nennen Handlungsempfehlungen, wie diese potenziellen Nutzer auf Basis der Erkenntnisse bei der Budgetallokation in der Instandhaltung modularer Systeme vorgehen sollten.

Besondere Relevanz besitzen die Erkenntnisse dieser Arbeit für Angehörige von Abteilungen in Unternehmen, Institutionen und öffentlichen Einrichtungen, die dort für Budgetallokation im Bereich der Instandhaltung verantwortlich sind.

Insbesondere bei der Vorgabe eines Gesamtbudgets für mehrere Perioden (z. B. Jahre oder Quartale), welches für die Instandhaltung eines modular aufgebauten Systems (wie bspw.

---

<sup>1</sup>Weitere Probleme, die sich konkret auf den DDPG-Algorithmus beziehen, insbesondere wenn die Belohnungen meist gleich Null sind, finden sich in Matheron u. a. (2019).

ein Straßennetzwerk) zur Verfügung steht, können die Erkenntnisse der Arbeit helfen, dieses Budget besonders effizient einzusetzen. Dafür sollten allerdings die beschriebenen Voraussetzungen gegeben sein. Das bedeutet insbesondere, dass die Planungszielsetzung „Maximierung der minimalen Periodensystemleistung“ übereinstimmen muss, dass die Zustandsbeschreibung der Komponenten als prozentuale Abnutzungsvorräte oder Zuverlässigkeiten gegeben sein muss, dass Übergangs- und Strukturbewertungsfunktionen ermittelt werden können oder dass Instandhaltungsdauern vernachlässigt werden können.

Ist aus Erfahrungswerten bekannt, dass keine allzu große Varianz der Verschleißprozesse der Komponenten des Systems vorliegt, kann mit Erwartungswerten für den Verschleiß gearbeitet werden und sehr schnell über einen mathematischen Solver wie CONOPT3<sup>2</sup> eine Empfehlung für die Budgetallokation für alle Perioden im Sinne einer starren Planung gegeben werden. Für die erste Periode kann das entsprechende Geld direkt freigegeben werden.

Vor dem Beginn jeder weiteren Periode des Planungszeitraums sollte, wenn möglich, eine Zustandsüberprüfung der einzelnen Komponenten stattfinden und im Sinne einer rollierenden Planung erneut für den restlichen Planungszeitraum eine Optimierung durchgeführt werden. Dabei wird Geld entsprechend immer nur für die direkt folgende Periode freigegeben.

Da die Varianz der Verschleißprozesse von dem jeweiligen Anwendungsfall in der Praxis abhängt, kann vor allem bei hoher Varianz bzw. großer Abweichung vom maximal möglichen Zielfunktionswert (vgl. Abschnitt 7.3.3) der Einsatz von flexiblen Planungsmethoden wie der behandelten (Deep-)Reinforcement-Learning-Methode „DDPG“ die Effizienz des Einsatzes der finanziellen Mittel verbessern. Dazu muss die zugrundeliegende Wahrscheinlichkeitsverteilung entweder bekannt sein oder über Simulation nachgebildet werden können. Auch eine empirische Verteilung, die stets aus historischen Daten ermittelt werden kann, kann hier ohne Weiteres verwendet werden. Ist dann das neuronale Netz einmal trainiert, kann eine zustandsabhängige Budgetallokation ad hoc vorgenommen werden. Für größere Instanzen sollte das Training nicht auf einer CPU eines handelsüblichen Rechners vorgenommen, sondern auf ein oder mehreren GPUs gerechnet werden.

---

<sup>2</sup>Für sehr große Systeme und/oder sehr lange Planungszeiträume kann es auch sinnvoll sein, den weniger ausgereiften, aber auf größere Instanzen ausgelegten Solver CONOPT4 (s. Drud (2021a)) zu verwenden.

## 8.4 Forschungsbedarf

Im vorliegenden Abschnitt sollen offene Fragen angesprochen werden, die sich im Laufe der Analyse ergeben haben. Wir gliedern diese Fragen wie folgt:

1. In dieser Arbeit besprochene Modelle und Methoden,
2. Änderungen an den mathematischen Modellen,
3. Forschungsbedarf zu alternativen Methoden und
4. Anwendung in der Praxis.

**In dieser Arbeit besprochene Modelle und Methoden.** Es kann tiefergehend untersucht werden, ob die Grenznutzenmethode auch auf den mehrperiodigen Planungszeitraum erweiterbar ist. Offen ist zudem die Frage, ob die Komposition der mehrdimensionalen monoton wachsenden Strukturbewertungsfunktion mit den konkaven Zustandsübergangsfunktionen quasikonkav ist, insbesondere wenn sich die Form der Strukturbewertungsfunktion an die Form einer Strukturfunktion (vgl. Beichelt (1993, S. 57 ff.)) anlehnt. Folgende Quellen könnten bei der Beantwortung dieser Frage hilfreich sein: Avriel u. a. (2010); Boyd u. a. (2009); Hadjisavvas u. a. (2005).

**Änderungen an den mathematischen Modellen.** Es wurde nicht untersucht, wie sich Änderungen an den mathematischen Modellen der Kapitel 4 oder 6 auswirken. Insbesondere die Untersuchung von alternativen Zielfunktionen bedarf einer genaueren Betrachtung. Inwieweit lassen sich die bereits geführten Beweise auf die geänderten Modelle übertragen? Gibt es mehrere Kandidaten für globale Optima? Des Weiteren wurden in Abschnitt 2.3 eine Menge von Abhängigkeiten insbesondere aus der Arbeit von Olde Keizer u. a. (2017) angesprochen. Wie lassen sich solche Abhängigkeiten über die Budgetbeschränkung hinaus in die bestehende Modellierung integrieren?

**Forschungsbedarf zu alternativen Methoden.** Für alternative Zielfunktionen, die andere mathematische Eigenschaften besitzen als die in dieser Arbeit behandelten, könnte das Konzept der Importanz einzelner Komponenten noch an Bedeutung gewinnen. Ausgangspunkt hierfür könnten die Überlegungen zur Grenznutzenmethode des Abschnitts 5.3.2 sein. Eine weitere offene Frage ist, ob neuere Methoden des sich aktuell rasant weiterentwickelnden Deep Reinforcement Learning noch leistungsfähiger als der vorgestellte DDPG-Algorithmus sein können. Dies könnte beispielsweise die Nutzung vortrainierter neuronaler Netze (s. Anderson u. a. (2015)) oder die Priorisierung der

Entnahme aus dem Batch (s. Barth-Maron u. a. (2018)) betreffen. Seit kurzem stellt das Acme-Framework (s. Hoffman u. a. (2020)) der Firma DeepMind Programmcode für vielversprechende und teilweise noch nicht veröffentlichte Algorithmen bereit. Ein weiteres wachsendes Gebiet, das insbesondere für modulare Systeme interessant sein könnte, ist das des „Multi-Agent Reinforcement Learning“ („MARL“, s. Busoniu u. a. (2008)). Man könnte sich anstelle einer gesamthaften Modellierung jede einzelne Komponente als eigenen Agenten vorstellen, der versucht, sich über Verhandlungen mit anderen Komponenten selbst zu optimieren. Insbesondere für stetige Aktions- und Zustandsräume scheint dies ein interessantes und noch wenig bearbeitetes Forschungsfeld zu sein.

**Anwendung in der Praxis.** Nicht betrachtet wurde in dieser Arbeit die Anwendbarkeit in der Praxis. Typische Anwendungsfälle sind die Instandhaltung von Straßen- oder Brückennetzwerken. Lässt sich die Theorie hier ohne Weiteres auf die Praxis übertragen? Wie formuliert man die Strukturbewertungsfunktion? Wie wählt man bspw. die Parameter der Investitionsfunktionen? Wie integriert man weitere anwendungsspezifische Nebenbedingungen? Dies sind spannende Fragen, die es zu untersuchen lohnt.

# 9 Zusammenfassung

---

Im Mittelpunkt der Arbeit stand die Frage, wie die Budgetallokation in der Instandhaltung modularer Systeme am besten erfolgen kann, wenn ein Budget periodenübergreifend zur Verfügung steht und nichtstationäre Verschleißprozesse der einzelnen Komponenten des Systems vorliegen. Es wurden dazu zwei verschiedene Planungsansätze vorgestellt und verglichen, die als Zielsetzung haben, die minimale Perioden-Systemleistung zu maximieren.

Es sollten insbesondere dahingehend Erkenntnisse gewonnen werden, wie diese beiden Planungsansätze – rollierende Planung und flexible Planung – die Investitionshöhe in Instandhaltungsmaßnahmen einzelner Komponenten und Perioden bestmöglich festlegen.

Für das Thema der Dissertation zentral waren die Begriffe Budgetallokation, Instandhaltung, modulare Systeme und Verschleiß. Daher widmete sich Kapitel 2 zunächst den Definitionen und Erläuterungen dieser begrifflichen Grundlagen und begann mit einem Überblick über die Grundmaßnahmen der Instandhaltung sowie über Instandhaltungsarten. Im Rahmen der Budgetallokation wurde insbesondere deren Wirkung im Kontext des abnehmenden Grenznutzens finanzieller Mittel besprochen. Für modulare Systeme wurden Darstellungsmöglichkeiten und Typen von Abhängigkeiten zwischen den einzelnen Komponenten des Systems behandelt. Weiterhin wurden die in dieser Arbeit verwendete stochastische Verteilung der Verschleißprozesse, die Gamma-Verteilung, im Kontext der Instandhaltung motiviert. Vor diesem Hintergrund wurden im Anschluss das Planungsproblem formuliert und sinnvolle Ansätze zur Lösung des Planungsproblems besprochen: die rollierende Planung und die flexible Planung mit neuronalen Netzen.

Die Analyse der Literatur, dargestellt in Kapitel 3, ergab Folgendes: Die untersuchten Arbeiten beschäftigen sich mit der Budgetallokation bei 1) monolithischen, mehrperiodigen, 2) modularen, einperiodigen und 3) modularen, mehrperiodigen Instandhaltungsmodellen. Besonders interessant war die Erkenntnis, dass es nur wenige Arbeiten gab, die sich



bei modularen, mehrperiodigen Instandhaltungsmodellen ein Gesamtbudget für alle Perioden vorgegeben haben („periodenübergreifende Budgetallokation“). Als Forschungslücke wurde identifiziert, dass bei diesen Arbeiten einige Aspekte nicht ausreichend Beachtung finden. So wurden vor allem nichtstationäre Verschleißprozesse im Voraus von Betrachtungen ausgeschlossen, obwohl solche Verschleißprozesse in der Praxis vorliegen können.

Es wurden für die beiden Planungsansätze zwei mathematische Modelle im Stil von dynamischen Optimierungsproblemen aufgestellt und methodisch unterschiedlich behandelt. Das Erwartungswertmodell, das Grundlage für die rollierende Planung war, wurde in Kapitel 4 entwickelt. Es wurden mit der Maximierung der Perioden-Systemleistung die Zielsetzung sowie die Bewertung der vorliegenden Systemstruktur und die Dynamik des Modells über Zustandsübergangsfunktionen modelliert, wobei letztere den abnehmenden Grenznutzen von Investitionen in Instandhaltungsmaßnahmen berücksichtigen.

In Kapitel 5 wurden mehrere Analysen des Erwartungswertmodells durchgeführt: Es wurde gezeigt, dass es stets sinnvoll ist, das gesamte Budget einzusetzen, es wurde eine geeignete Konkretisierung der Zustandsübergangsfunktion mit Hilfe einer nichtlinearen Differentialgleichung hergeleitet und für diese eine abfallende Struktur der Perioden-Systemleistungen im Zeitverlauf für die optimale Lösung nachgewiesen. Zudem wurde für das einperiodige Planungsproblem ein Grenznutzen-Algorithmus entwickelt, der eine Lösung findet, die notwendigen Optimalitätsbedingungen erster Ordnung genügt. Über die Komposition von Strukturbewertungs- und Zustandsübergangsfunktion bzw. Systemleistungs- und Zielfunktion wurde eine effiziente Lösbarkeit für quasikonkave Spezialfälle der Strukturbewertungsfunktion im mehrperiodigen Fall bewiesen.

Im folgenden Kapitel 6 wurde für die flexible Planung mit neuronalen Netzen eine Modellierung als stochastisches dynamisches Programm gewählt und mit einer Methode aus dem Bereich des Deep Reinforcement Learning gelöst: dem Actor-Critic-Algorithmus „Deep Deterministic Policy Gradient“ (DDPG). Dieses Verfahren wurde gewählt, da es spezifisch auf den Umgang mit überabzählbaren Zustands- und Aktionsräumen ausgelegt ist, die in der Modellierung des Problems angenommen werden.

Über Ergebnisse experimenteller Performance-Analysen der zuvor entwickelten und vorbereiteten Planungsansätze gibt Kapitel 7 Auskunft. Es wurde gezeigt, dass eine periodenübergreifende Budgetallokation im Vergleich zu einer periodenbezogenen deutliche Vorteile bezüglich der Zielsetzung hat. Weiterhin konnte die Effizienz und Genauigkeit der

Grenznutzenmethode im Vergleich mit kommerziellen Solvern für nichtlineare Programme belegt werden. Im Anschluss wurde der Deep-Reinforcement-Learning-Ansatz mit einer Planungsmethode der rollierenden Planung und einer zufälligen Budgetallokation verglichen. Die untersuchten Instanzen wurden dabei selbst generiert, da in der Literatur kein vergleichbares Datenset gefunden werden konnte. Die Implementierungen erfolgten mit der Modellierungssprache GAMS bzw. in der Programmiersprache Python. Es konnte gezeigt werden, dass beide vorgestellten Planungsansätze deutlich besser als eine zufällige Planung sind. Über eine Lasso-Regression wurde untersucht, welche Einflussgrößen den einen oder den anderen Planungsansatz begünstigen.

In Kapitel 8 wird dargelegt, wie die vorgestellten Planungsansätze nichtstationäre Verschleißprozesse im Kontext modularer Systeme in der Instandhaltung handhaben und ihre Anwendungsberechtigung auch in der Praxis haben können, insbesondere in der taktischen Planung. In der Zukunft ist im Themengebiet zu erwarten, dass es vor allem weitere Fortschritte bei der Entwicklung von Algorithmen des Deep Reinforcement Learning geben wird. Bessere Hardware wird zusätzlich dazu beitragen, dass auch große Instanzen in schneller Zeit trainiert und gelöst werden können. Da das Konzept des Reinforcement Learning universell einsetzbar ist, ist anzunehmen, dass auch weitere Forschungsfelder die Anwendbarkeit für sich entdecken werden.

# Literaturverzeichnis

- [Agrawal und Boyd 2020] AGRAWAL, Akshay ; BOYD, Stephen: Disciplined quasiconvex programming. In: *Optimization Letters* 14 (2020), Nr. 7, 1643–1657. <http://dx.doi.org/10.1007/s11590-020-01561-8>. – ISSN 1862–4472
- [Alaswad und Xiang 2017] ALASWAD, Suzan ; XIANG, Yisha: A review on condition-based maintenance optimization models for stochastically deteriorating system. In: *Reliability Engineering & System Safety* 157 (2017), 54–63. <http://dx.doi.org/10.1016/j.ress.2016.08.009>. – ISSN 09518320
- [Ameri u. a. 2019] AMERI, Mahmoud ; JARRAHI, Armin ; HADDADI, Farshad ; MIRABI-MOGHADDAM, Mohammad H.: A Two-Stage Stochastic Model for Maintenance and Rehabilitation Planning of Pavements. In: *Mathematical Problems in Engineering* 2019 (2019), S. 1–15. – ISSN 1024–123X
- [American Society of Civil Engineers (ASCE) 2021] AMERICAN SOCIETY OF CIVIL ENGINEERS (ASCE): *Report Card for America’s Infrastructure*. <https://infrastructurereportcard.org/cat-item/roads>. (Abrufdatum: 2021-04-19). 2021
- [Anderson u. a. 2015] ANDERSON, Charles W. ; LEE, Minwoo ; ELLIOTT, Daniel L.: Faster reinforcement learning after pretraining deep networks to predict state dynamics. In: *2015 International Joint Conference on Neural Networks (IJCNN)* Bd. 2015-Sept, IEEE, 2015. – ISBN 978–1–4799–1960–4, 1–7
- [Andrei 2017a] ANDREI, Neculai: *Springer Optimization and Its Applications*. Bd. 121: *Continuous Nonlinear Optimization for Engineering Applications in GAMS Technology*. Cham : Springer International Publishing, 2017. <http://dx.doi.org/10.1007/978-3-319-58356-3>. – ISBN 978–3–319–58355–6
- [Andrei 2017b] ANDREI, Neculai: Generalized Reduced Gradient with Sequential Linearization: CONOPT. 2017. 2017. [http://dx.doi.org/10.1007/978-3-319-58356-3\\_16](http://dx.doi.org/10.1007/978-3-319-58356-3_16). – ISBN 978–3–319–58355–6, S. 331–342

- [Arrow und Enthoven 1961] ARROW, Kenneth J. ; ENTHOVEN, Alain C.: Quasi-Concave Programming. In: *Econometrica* 29 (1961), Nr. 4, 779. <http://dx.doi.org/10.2307/1911819>. – ISSN 00129682
- [Avriel u. a. 2010] AVRIEL, Mordecai ; DIEWERT, Walter E. ; SCHAIBLE, Siegfried ; ZANG, Israel: *Generalized Concavity*. Society for Industrial and Applied Mathematics, 2010. <http://dx.doi.org/10.1137/1.9780898719437>. – ISBN 978-0-89871-896-6
- [Barlow und Proschan 1966] BARLOW, Richard E. ; PROSCHAN, Frank: *Mathematical Theory of Reliability*. Philadelphia : SIAM, 1966. – 258 S. – ISBN 0-89871-369-2\*pbk.
- [Barth-Maron u. a. 2018] BARTH-MARON, Gabriel ; HOFFMAN, Matthew W. ; BUDDEN, David ; DABNEY, Will ; HORGAN, Dan ; TB, Dhruva ; MULDAL, Alistair ; HEES, Nicolas ; LILLICRAP, Timothy: Distributed Distributional Deterministic Policy Gradients. In: *6th International Conference on Learning Representations, ICLR 2018 - Conference Track Proceedings* (2018), 1–16. <http://arxiv.org/abs/1804.08617>. – ISSN 23318422
- [Bazaraa u. a. 2006] BAZARAA, Mokhtar S. ; SHERALI, Hanif D. ; SHETTY, C. M.: *Nonlinear Programming*. 2. Hoboken, NJ, USA : John Wiley & Sons, Inc., 2006. – 871 S. <http://dx.doi.org/10.1002/0471787779>. – ISBN 9780471787778
- [Beichelt 1993] BEICHELT, Frank: *Zuverlässigkeits- und Instandhaltungstheorie*. Wiesbaden : Vieweg+Teubner Verlag, 1993. – 312 S. <http://dx.doi.org/10.1007/978-3-663-11921-0>. – ISBN 978-3-519-02985-4
- [Beichelt und Tittmann 2012] BEICHELT, Frank ; TITTMANN, Peter: *Reliability and Maintenance - Networks and Systems*. Boca Raton, FL, USA : Taylor & Francis, 2012. – ISBN 978-1-4398-2636-2
- [Bellman 1966] BELLMAN, Richard: Dynamic Programming. In: *Science* 153 (1966), Nr. 3731, 34–37. <http://dx.doi.org/10.1126/science.153.3731.34>. – ISBN 0-486-42809-5
- [Boyd u. a. 2009] BOYD, Stephen ; VANDENBERGHE, L. ; LIEVEN VANDENBERGHE: *Convex Optimization*. 7. Cambridge, 2009
- [Bundesministerium für Verkehr und digitale Infrastruktur (BMVI) 2020] BUNDESMINISTERIUM FÜR VERKEHR UND DIGITALE INFRASTRUKTUR (BMVI): Verkehrsinvestitionsbericht für das Berichtsjahr 2018. 2020. <https://www.bmvi.de/goto?id=469272>. 2020. – Bericht

- [Busoniu u. a. 2008] BUSONI, Lucian ; BABUSKA, Robert ; DE SCHUTTER, Bart: A Comprehensive Survey of Multiagent Reinforcement Learning. In: *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)* 38 (2008), Nr. 2, 156–172. <http://dx.doi.org/10.1109/TSMCC.2007.913919>. – ISSN 1094–6977
- [Cao u. a. 2018] CAO, Wenbin ; JIA, Xisheng ; HU, Qiwei ; ZHAO, Jianmin ; WU, Yutao: A literature review on selective maintenance for multi-unit systems. In: *Quality and Reliability Engineering International* 34 (2018), Nr. 5, 824–845. <http://dx.doi.org/10.1002/qre.2293>. – ISSN 07488017
- [Carpenter 1966] CARPENTER, William F.: On the Solution of the Real Quartic. In: *Mathematics Magazine* 39 (1966), Nr. 1, 28. <http://dx.doi.org/10.2307/2688990>. – ISSN 0025570X
- [Chan u. a. 2003] CHAN, Weng T. ; FWA, T. F. ; TAN, J. Y.: Optimal Fund-Allocation Analysis for Multidistrict Highway Agencies. In: *Journal of Infrastructure Systems* 9 (2003), Nr. 4, 167–175. [http://dx.doi.org/10.1061/\(ASCE\)1076-0342\(2003\)9:4\(167\)](http://dx.doi.org/10.1061/(ASCE)1076-0342(2003)9:4(167)). – ISSN 1076–0342
- [Cho und Parlar 1991] CHO, Danny I. D. ; PARLAR, Mahmut: A survey of maintenance models for multi-unit systems. In: *European Journal of Operational Research* 51 (1991), Nr. 1, 1–23. [http://dx.doi.org/10.1016/0377-2217\(91\)90141-H](http://dx.doi.org/10.1016/0377-2217(91)90141-H). – ISBN 0377–2217
- [Chootinan u. a. 2006] CHOOTINAN, Piya ; CHEN, Anthony ; HORROCKS, Matthew R. ; BOLLING, Doyt: A multi-year pavement maintenance program using a stochastic simulation-based genetic algorithm approach. In: *Transportation Research Part A: Policy and Practice* 40 (2006), Nr. 9, 725–743. <http://dx.doi.org/10.1016/j.tra.2005.12.003>. – ISSN 09658564
- [Chowdhury und Tan 2005] CHOWDHURY, Mashrur ; TAN, Paulin: Investment Analysis Using the Constraint Multiobjective Programming Method: A Case Study. In: *Transportation Research Record: Journal of the Transportation Research Board* 1924 (2005), 231–232. <http://dx.doi.org/10.3141/1924-29>. – ISSN 0361–1981
- [Clarivate Analytics 2020] CLARIVATE ANALYTICS: *Web of Science Core Collection on Help - Wildcards*. <https://images.webofknowledge.com/WOKRS59B4{ }2/help/WOS/hs{ }wildcards.html>. (Abrufdatum: 2021-04-19). 2020

- [Dehghani u. a. 2013] DEHGHANI, Mohammad S. ; GIUSTOZZI, Filippo ; FLINTSCH, Gerardo W. ; CRISPINO, Maurizio: Cross-Asset Resource Allocation Framework for Achieving Performance Sustainability. In: *Transportation Research Record: Journal of the Transportation Research Board* 2361 (2013), Nr. 1, 16–24. <http://dx.doi.org/10.3141/2361-03>. – ISSN 0361–1981
- [Dehghani und Sherali 2016] DEHGHANI, Mohammad S. ; SHERALI, Hanif D.: A resource allocation approach for managing critical network-based infrastructure systems. In: *IEEE Transactions* 48 (2016), Nr. 9, 826–837. <http://dx.doi.org/10.1080/0740817X.2016.1147662>. – ISSN 0740–817X
- [Dekker u. a. 1997] DEKKER, Rommert ; WILDEMAN, Ralph E. ; VAN DER DUYN SCHOUTEN, Frank A.: A review of multi-component maintenance models with economic dependence. In: *Mathematical Methods of Operations Research* 45 (1997), Nr. 3, 411–435. <http://dx.doi.org/10.1007/BF01194788>. – ISBN 978–0–415–44786–7
- [DIN-Normen-Ausschuss Dienstleistungen (NADL) 2019] DIN-NORMEN-AUSSCHUSS DIENSTLEISTUNGEN (NADL): *DIN 31051:2019-06, Grundlagen der Instandhaltung*. 2019
- [DIN-Normenausschuss Dienstleistungen (NADL) 2018] DIN-NORMENAUSSCHUSS DIENSTLEISTUNGEN (NADL): *DIN EN 13306:2018-02, Instandhaltung - Begriffe der Instandhaltung; Dreisprachige Fassung EN 13306:2017*. 2018
- [Ding und Kamaruddin 2015] DING, Siew-Hong ; KAMARUDDIN, Shahrul: Maintenance policy optimization - literature review and directions. In: *The International Journal of Advanced Manufacturing Technology* 76 (2015), Nr. 5-8, 1263–1283. <http://dx.doi.org/10.1007/s00170-014-6341-2>. – ISBN 0017001463
- [Drud 2020] DRUD, Arne: *CONOPT Optimization Solver*. [https://www.gams.com/latest/docs/S\\_{\\_}CONOPT.html](https://www.gams.com/latest/docs/S_{_}CONOPT.html). (Abrufdatum: 2020-11-30). 2020
- [Drud 2021a] DRUD, Arne: *CONOPT4*. [https://www.gams.com/latest/docs/S\\_{\\_}CONOPT4.html](https://www.gams.com/latest/docs/S_{_}CONOPT4.html). (Abrufdatum: 2021-02-26). 2021
- [Drud 2021b] DRUD, Arne: *GAMS documentation for CONOPT - Smooth Approximations of MIN and MAX*. [https://www.gams.com/latest/docs/S\\_{\\_}CONOPT.html{#}CONOPT\\_{\\_}SMOOTH\\_{\\_}APPROXIMATIONS](https://www.gams.com/latest/docs/S_{_}CONOPT.html{#}CONOPT_{_}SMOOTH_{_}APPROXIMATIONS). (Abrufdatum: 2021-04-19). 2021
- [Ellingwood und Mori 1993] ELLINGWOOD, Bruce R. ; MORI, Yasuhiro: Probabilistic methods for condition assessment and life prediction of concrete structures in nuclear

- power plants. In: *Nuclear Engineering and Design* 142 (1993), Nr. 2-3, 155–166. [http://dx.doi.org/10.1016/0029-5493\(93\)90199-J](http://dx.doi.org/10.1016/0029-5493(93)90199-J). – ISSN 00295493
- [Feess und Gillenkirch 2018] FEESS, Eberhard ; GILLENKIRCH, Robert: *Definition System*. <https://wirtschaftslexikon.gabler.de/definition/system-50117/version-273342>. (Abrufdatum: 2020-01-15). 2018
- [Forster 2016] FORSTER, Otto: *Analysis 1*. Bd. 12. 12. Wiesbaden : Springer Fachmedien Wiesbaden, 2016. – 1–348 S. <http://dx.doi.org/10.1007/978-3-658-11545-6>. – ISBN 978-3-658-11544-9
- [Forster 2017] FORSTER, Otto: *Analysis 2*. 11. Wiesbaden : Springer Fachmedien Wiesbaden, 2017 (Springer-Lehrbuch). <http://dx.doi.org/10.1007/978-3-658-19411-6>. – ISBN 978-3-658-19410-9
- [France-Mensah und O’Brien 2018] FRANCE-MENSAH, Jojo ; O’BRIEN, William J.: Budget Allocation Models for Pavement Maintenance and Rehabilitation: Comparative Case Study. In: *Journal of Management in Engineering* 34 (2018), Nr. 2, 05018002. [http://dx.doi.org/10.1061/\(ASCE\)ME.1943-5479.0000599](http://dx.doi.org/10.1061/(ASCE)ME.1943-5479.0000599). – ISSN 0742-597X
- [Fwa u. a. 2000] FWA, T. F. ; CHAN, W. T. ; HOQUE, K. Z.: Multiobjective Optimization for Pavement Maintenance Programming. In: *Journal of Transportation Engineering* 126 (2000), Nr. 5, 367–374. [http://dx.doi.org/10.1061/\(ASCE\)0733-947X\(2000\)126:5\(367\)](http://dx.doi.org/10.1061/(ASCE)0733-947X(2000)126:5(367)). – ISSN 0733-947X
- [Fwa und Farhan 2012] FWA, T. F. ; FARHAN, J.: Optimal Multiasset Maintenance Budget Allocation in Highway Asset Management. In: *Journal of Transportation Engineering* 138 (2012), Nr. 10, 1179–1187. [http://dx.doi.org/10.1061/\(ASCE\)TE.1943-5436.0000414](http://dx.doi.org/10.1061/(ASCE)TE.1943-5436.0000414). – ISSN 0733-947X
- [Gabriel u. a. 2006] GABRIEL, Steven A. ; ORDÓÑEZ, Javier F. ; FARIA, José A.: Contingency Planning in Project Selection Using Multiobjective Optimization and Chance Constraints. In: *Journal of Infrastructure Systems* 12 (2006), Nr. 2, 112–120. [http://dx.doi.org/10.1061/\(ASCE\)1076-0342\(2006\)12:2\(112\)](http://dx.doi.org/10.1061/(ASCE)1076-0342(2006)12:2(112)). – ISSN 1076-0342
- [GAMS Development Corporation 2020] GAMS DEVELOPMENT CORPORATION: *General Algebraic Modeling System (GAMS)*. <https://www.gams.com/download/>. 2020
- [GAMS Development Corporation 2021] GAMS DEVELOPMENT CORPORATION: *GAMS*

- Python API*. <https://www.gams.com/31/docs/API{ }PY{ }OVERVIEW.html>. (Abrufdatum: 2021-04-19). 2021
- [Gao u. a. 2012] GAO, Lu ; XIE, Chi ; ZHANG, Zhanmin ; WALLER, S. T.: Network-Level Road Pavement Maintenance and Rehabilitation Scheduling for Optimal Performance Improvement and Budget Utilization. In: *Computer-Aided Civil and Infrastructure Engineering* 27 (2012), Nr. 4, 278–287. <http://dx.doi.org/10.1111/j.1467-8667.2011.00733.x>. – ISSN 10939687
- [Gao und Zhang 2013] GAO, Lu ; ZHANG, Zhanmin: Management of Pavement Maintenance, Rehabilitation, and Reconstruction through Network Partition. In: *Transportation Research Record: Journal of the Transportation Research Board* 2366 (2013), Nr. 1, 59–63. <http://dx.doi.org/10.3141/2366-07>. – ISSN 0361–1981
- [de la Garza u. a. 2011] GARZA, Jesus M. I. ; AKYILDIZ, Sercan ; BISH, Doug R. ; KRUEGER, Denise A.: Network-level optimization of pavement maintenance renewal strategies. In: *Advanced Engineering Informatics* 25 (2011), Nr. 4, 699–712. <http://dx.doi.org/10.1016/j.aei.2011.08.002>. – ISBN 1474–0346
- [Gharaibeh u. a. 2006] GHARAIBEH, Nasir G. ; CHIU, Yi-Chang ; GURIAN, Patrick L.: Decision Methodology for Allocating Funds across Transportation Infrastructure Assets. In: *Journal of Infrastructure Systems* 12 (2006), Nr. 1, 1–9. [http://dx.doi.org/10.1061/\(ASCE\)1076-0342\(2006\)12:1\(1\)](http://dx.doi.org/10.1061/(ASCE)1076-0342(2006)12:1(1)). – ISBN 1920017046
- [Golabi u. a. 1982] GOLABI, K. ; KULKARNI, R. B. ; WAY, G. B.: A Statewide Pavement Management System. In: *Interfaces* 12 (1982), dec, Nr. 6, 5–21. <http://dx.doi.org/10.1287/inte.12.6.5>. – ISBN 0891243208
- [Golabi und Shepard 1997] GOLABI, Kamal ; SHEPARD, Richard: Pontis: A System for Maintenance Optimization and Improvement of US Bridge Networks. In: *Interfaces* 27 (1997), Nr. 1, 71–88. <http://dx.doi.org/10.1287/inte.27.1.71>. – ISSN 0092–2102
- [Golub u. a. 1999] GOLUB, Gene H. ; HANSEN, Per C. ; O’LEARY, Dianne P.: Tikhonov Regularization and Total Least Squares. In: *SIAM Journal on Matrix Analysis and Applications* 21 (1999), Nr. 1, 185–194. <http://dx.doi.org/10.1137/S0895479897326432>. – ISSN 0895–4798
- [Gorjian u. a. 2010] GORJIAN, Nima ; MA, Lin ; MITTINTY, Murthy ; YARLAGADDA, Prasad ; SUN, Yong: A review on degradation models in reliability analysis. 2010. In:



- Engineering Asset Lifecycle Management*. London : Springer London, 2010 ( September). [http://dx.doi.org/10.1007/978-0-85729-320-6\\_42](http://dx.doi.org/10.1007/978-0-85729-320-6_42). – ISBN 9781849960021, 369–384
- [Grussing und Liu 2014] GRUSSING, Michael N. ; LIU, Liang Y.: Knowledge-Based Optimization of Building Maintenance, Repair, and Renovation Activities to Improve Facility Life Cycle Investments. In: *Journal of Performance of Constructed Facilities* 28 (2014), Nr. 3, 539–548. [http://dx.doi.org/10.1061/\(ASCE\)CF.1943-5509.0000449](http://dx.doi.org/10.1061/(ASCE)CF.1943-5509.0000449). – ISSN 0887–3828
- [Hackl u. a. 2018] HACKL, Jürgen ; ADEY, Bryan T. ; LETHANH, Nam: Determination of Near-Optimal Restoration Programs for Transportation Networks Following Natural Hazard Events Using Simulated Annealing. In: *Computer-Aided Civil and Infrastructure Engineering* 33 (2018), Nr. 8, 618–637. <http://dx.doi.org/10.1111/mice.12346>. – ISSN 10939687
- [Hadjisavvas u. a. 2005] HADJISAVVAS, Nicolas (Hrsg.) ; KOMLÓSI, Sándor (Hrsg.) ; SCHAIBLE, Siegfried (Hrsg.): *Nonconvex Optimization and Its Applications*. Bd. 76: *Handbook of Generalized Convexity and Generalized Monotonicity*. Boston : Kluwer Academic Publishers, 2005. <http://dx.doi.org/10.1007/b101428>. – ISBN 0–387–23255–9
- [Hastie u. a. 2007] HASTIE, Trevor ; TAYLOR, Jonathan ; TIBSHIRANI, Robert ; WALTHER, Guenther: Forward stagewise regression and the monotone lasso. In: *Electronic Journal of Statistics* 1 (2007), 1–29. <http://dx.doi.org/10.1214/07-EJS004>. – ISSN 1935–7524
- [Hessel u. a. 2018] HESSEL, Matteo ; MODAYIL, Joseph ; VAN HASSELT, Hado ; SCHAUL, Tom ; OSTROVSKI, Georg ; DABNEY, Will ; HORGAN, Dan ; PIOT, Bilal ; AZAR, Mohammad ; SILVER, David: Rainbow: Combining improvements in deep reinforcement learning. In: *32nd AAAI Conference on Artificial Intelligence, AAAI 2018* (2018), S. 3215–3222. ISBN 9781577358008
- [Hoffman u. a. 2020] HOFFMAN, Matt ; SHAHRIARI, Bobak ; ASLANIDES, John ; BARTH-MARON, Gabriel ; BEHBAHANI, Feryal ; NORMAN, Tamara ; ABDOLMALEKI, Abbas ; CASSIRER, Albin ; YANG, Fan ; BAUMLI, Kate ; HENDERSON, Sarah ; NOVIKOV, Alex ; COLMENAREJO, Sergio G. ; CABI, Serkan ; GULCEHRE, Caglar ; PAINE, Tom L. ; COWIE, Andrew ; WANG, Ziyu ; PIOT, Bilal ; FREITAS, Nando de: Acme: A

- Research Framework for Distributed Reinforcement Learning. In: *arXiv* (2020), 1–33. <http://arxiv.org/abs/2006.00979>. – ISSN 23318422
- [Kang u. a. 2019] KANG, Jichuan ; SOBRAL, Jose ; SOARES, C. G.: Review of Condition-Based Maintenance Strategies for Offshore Wind Energy. In: *Journal of Marine Science and Application* 18 (2019), Nr. 1, 1–16. <http://dx.doi.org/10.1007/s11804-019-00080-y>. – ISSN 1671–9433
- [Kingma und Ba 2014] KINGMA, Diederik P. ; BA, Jimmy: Adam: A Method for Stochastic Optimization. (2014), dec. <http://arxiv.org/abs/1412.6980>
- [Koschnick 1996] KOSCHNICK, Wolfgang J.: *Management - Enzyklopädisches Lexikon*. Berlin, Boston : De Gruyter, 1996. <http://dx.doi.org/10.1515/9783110818987>. – ISBN 9783110818987
- [Kuhn 2010] KUHN, Kenneth D.: Network-Level Infrastructure Management Using Approximate Dynamic Programming. In: *Journal of Infrastructure Systems* 16 (2010), Nr. 2, 103–111. [http://dx.doi.org/10.1061/\(ASCE\)IS.1943-555X.0000019](http://dx.doi.org/10.1061/(ASCE)IS.1943-555X.0000019). – ISSN 1076–0342
- [Kuhn 2012] KUHN, Kenneth D.: Pavement Network Maintenance Optimization Considering Multidimensional Condition Data. In: *Journal of Infrastructure Systems* 18 (2012), dec, Nr. 4, 270–277. [http://dx.doi.org/10.1061/\(ASCE\)IS.1943-555X.0000077](http://dx.doi.org/10.1061/(ASCE)IS.1943-555X.0000077). – ISSN 1076–0342
- [Lee und Madanat 2015] LEE, Jinwoo ; MADANAT, Samer: A joint bottom-up solution methodology for system-level pavement rehabilitation and reconstruction. In: *Transportation Research Part B: Methodological* 78 (2015), 106–122. <http://dx.doi.org/10.1016/j.trb.2015.05.001>. – ISSN 01912615
- [Leidinger 2017] LEIDINGER, Bernhard: *Wertorientierte Instandhaltung*. Wiesbaden : Springer Fachmedien Wiesbaden, 2017. <http://dx.doi.org/10.1007/978-3-658-17855-0>. – ISBN 978–3–658–17854–3
- [Li und Madanat 2002] LI, Yuwei ; MADANAT, Samer: A steady-state solution for the optimal pavement resurfacing problem. In: *Transportation Research Part A: Policy and Practice* 36 (2002), Nr. 6, 525–535. [http://dx.doi.org/10.1016/S0965-8564\(01\)00020-9](http://dx.doi.org/10.1016/S0965-8564(01)00020-9). – ISSN 09658564
- [Lillicrap u. a. 2015] LILLICRAP, Timothy P. ; HUNT, Jonathan J. ; PRITZEL, Alexander ; HEES, Nicolas ; EREZ, Tom ; TASSA, Yuval ; SILVER, David ; WIERSTRA, Daan:

- Continuous control with deep reinforcement learning. In: *4th International Conference on Learning Representations, ICLR 2016 - Conference Track Proceedings* (2015). <http://arxiv.org/abs/1509.02971>
- [Lin 1993] LIN, Long-Ji: *Reinforcement learning for robots using neural networks*, Carnegie Mellon University, PhD Thesis, 1993. <https://isl.anthropomatik.kit.edu/pdf/Lin1993.pdf>. – 160 S.
- [Lisnianski u. a. 2010] LISNIANSKI, Anatoly ; FRENKEL, Ilia ; DING, Yi: *Multi-state system reliability analysis and optimization for engineers and industrial managers*. London : Springer London, 2010. – 1–393 S. <http://dx.doi.org/10.1007/978-1-84996-320-6>. – ISBN 978–1–84996–319–0
- [Liu und Frangopol 2005] LIU, Min ; FRANGOPOL, Dan M.: Multiobjective Maintenance Planning Optimization for Deteriorating Bridges Considering Condition, Safety, and Life-Cycle Cost. In: *Journal of Structural Engineering* 131 (2005), Nr. 5, 833–842. [http://dx.doi.org/10.1061/\(ASCE\)0733-9445\(2005\)131:5\(833\)](http://dx.doi.org/10.1061/(ASCE)0733-9445(2005)131:5(833)). – ISSN 0733–9445
- [Mangasarian 1969] MANGASARIAN, Olvi L.: *Nonlinear Programming*. SIAM, 1969
- [Matheron u. a. 2019] MATHERON, Guillaume ; PERRIN, Nicolas ; SIGAUD, Olivier: The problem with DDPG: understanding failures in deterministic environments with sparse rewards. In: *arXiv* (2019), 1–19. <http://arxiv.org/abs/1911.11679>. – ISSN 23318422
- [Medury und Madanat 2013] MEDURY, Aditya ; MADANAT, Samer: Incorporating network considerations into pavement management systems: A case for approximate dynamic programming. In: *Transportation Research Part C: Emerging Technologies* 33 (2013), 134–150. <http://dx.doi.org/10.1016/j.trc.2013.03.003>. – ISBN 9780888652454
- [Medury und Madanat 2014] MEDURY, Aditya ; MADANAT, Samer: Simultaneous Network Optimization Approach for Pavement Management Systems. In: *Journal of Infrastructure Systems* 20 (2014), Nr. 3, 04014010. [http://dx.doi.org/10.1061/\(ASCE\)IS.1943-555X.0000149](http://dx.doi.org/10.1061/(ASCE)IS.1943-555X.0000149). – ISSN 1076–0342
- [Meegoda u. a. 2009] MEEGODA, Jay N. ; JULIANO, Thomas M. ; TANG, Chi: Culvert Information Management System. In: *Transportation Research Record: Journal of the Transportation Research Board* 2108 (2009), Nr. 1, 3–12. <http://dx.doi.org/10.3141/2108-01>. – ISSN 0361–1981

- [Menendez u. a. 2013] MENENDEZ, Jose R. ; SIABIL, Salar Z. ; NARCISO, Paul ; GHARAIBEH, Nasir G.: Prioritizing Infrastructure Maintenance and Rehabilitation Activities under various Budgetary Scenarios. In: *Transportation Research Record: Journal of the Transportation Research Board* 2361 (2013), Nr. 1, 56–62. <http://dx.doi.org/10.3141/2361-07>. – ISSN 0361–1981
- [Mild und Salo 2009] MILD, Pekka ; SALO, Ahti: Combining a Multiattribute Value Function with an Optimization Model: An Application to Dynamic Resource Allocation for Infrastructure Maintenance. In: *Decision Analysis* 6 (2009), Nr. 3, 139–152. <http://dx.doi.org/10.1287/deca.1090.0143>. – ISSN 1545–8490
- [Mnih u. a. 2013] MNIH, Volodymyr ; KAVUKCUOGLU, Koray ; SILVER, David ; GRAVES, Alex ; ANTONOGLU, Ioannis ; WIERSTRA, Daan ; RIEDMILLER, Martin: Playing Atari with Deep Reinforcement Learning. (2013), dec, 1–9. <http://arxiv.org/abs/1312.5602>
- [Mnih u. a. 2015] MNIH, Volodymyr ; KAVUKCUOGLU, Koray ; SILVER, David ; RUSU, Andrei A. ; VENESS, Joel ; BELLEMARE, Marc G. ; GRAVES, Alex ; RIEDMILLER, Martin ; FIDJELAND, Andreas K. ; OSTROVSKI, Georg ; PETERSEN, Stig ; BEATTIE, Charles ; SADIK, Amir ; ANTONOGLU, Ioannis ; KING, Helen ; KUMARAN, Dharshan ; WIERSTRA, Daan ; LEGG, Shane ; HASSABIS, Demis: Human-level control through deep reinforcement learning. In: *Nature* 518 (2015), Nr. 7540, 529–533. <http://dx.doi.org/10.1038/nature14236>. – ISSN 0028–0836
- [Mohammadi u. a. 2019] MOHAMMADI, Alireza ; AMADOR-JIMENEZ, Luis ; NASIRI, Fuzhan: Review of asset management for metro systems: challenges and opportunities. In: *Transport Reviews* 39 (2019), Nr. 3, 309–326. <http://dx.doi.org/10.1080/01441647.2018.1470119>. – ISSN 0144–1647
- [Mohammadi u. a. 2020] MOHAMMADI, Alireza ; AMADOR-JIMENEZ, Luis ; NASIRI, Fuzhan: Reliable, Effective, and Sustainable Urban Railways: A Model for Optimal Planning and Asset Management. In: *Journal of Construction Engineering and Management* 146 (2020), Nr. 6, 04020057. [http://dx.doi.org/10.1061/\(ASCE\)CE.1943-7862.0001839](http://dx.doi.org/10.1061/(ASCE)CE.1943-7862.0001839). – ISSN 0733–9364
- [Morcoux und Lounis 2005] MORCOUS, G. ; LOUNIS, Z.: Maintenance optimization of infrastructure networks using genetic algorithms. In: *Automation in Construction*

- 14 (2005), Nr. 1, 129–142. <http://dx.doi.org/10.1016/j.autcon.2004.08.014>. – ISSN 09265805
- [Nahas und Nourelfath 2005] NAHAS, Nabil ; NOURELFATH, Mustapha: Ant system for reliability optimization of a series system with multiple-choice and budget constraints. In: *Reliability Engineering & System Safety* 87 (2005), Nr. 1, 1–12. <http://dx.doi.org/10.1016/j.res.2004.02.007>. – ISSN 09518320
- [Nelson 1982] NELSON, Wayne: *Applied Life Data Analysis*. Hoboken, NJ, USA : John Wiley & Sons, Inc., 1982 (Wiley Series in Probability and Statistics). <http://dx.doi.org/10.1002/0471725234>. – ISBN 9780471725237
- [Nicolai und Dekker 2008] NICOLAI, Robin P. ; DEKKER, Rommert: Optimal Maintenance of Multi-component Systems: A Review. 2008. In: *Complex System Maintenance Handbook* Bd. 8. London : Springer London, 2008. [http://dx.doi.org/10.1007/978-1-84800-011-7\\_11](http://dx.doi.org/10.1007/978-1-84800-011-7_11). – ISSN 2196999X, 263–286
- [van Noortwijk 2009] NOORTWIJK, J.M. van: A survey of the application of gamma processes in maintenance. In: *Reliability Engineering & System Safety* 94 (2009), Nr. 1, 2–21. <http://dx.doi.org/10.1016/j.res.2007.03.019>. – ISSN 09518320
- [Normenausschuß Materialprüfung (NMP) im DIN Deutsches Institut für Normung e.V. 1979] NORMENAUSSCHUSS MATERIALPRÜFUNG (NMP) IM DIN DEUTSCHES INSTITUT FÜR NORMUNG E.V., Normenausschuß Gleitlager (NGL) im DIN: *DIN 50320:1979-12, Verschleiß; Begriffe, Systemanalyse von Verschleißvorgängen, Gliederung des Verschleißgebietes*. 1979
- [Nozhati u. a. 2019] NOZHATI, Saeed ; SARKALE, Yugandhar ; ELLINGWOOD, Bruce ; CHONG, Edwin K. P. ; MAHMOUD, Hussam: Near-optimal planning using approximate dynamic programming to enhance post-hazard community resilience management. In: *Reliability Engineering & System Safety* 181 (2019), Nr. September 2018, 116–126. <http://dx.doi.org/10.1016/j.res.2018.09.011>. – ISSN 09518320
- [Nwankpa u. a. 2018] NWANKPA, Chigozie ; IJOMAH, Winifred ; GACHAGAN, Anthony ; MARSHALL, Stephen: Activation Functions: Comparison of trends in Practice and Research for Deep Learning. In: *arXiv* (2018), 1–20. <http://arxiv.org/abs/1811.03378>. – ISSN 23318422
- [Olde Keizer u. a. 2017] OLDE KEIZER, Minou C. ; FLAPPER, Simme Douwe P. ; TEUNTER, Ruud H.: Condition-based maintenance policies for systems with multiple dependent

- components: A review. In: *European Journal of Operational Research* 261 (2017), Nr. 2, 405–420. <http://dx.doi.org/10.1016/j.ejor.2017.02.044>. – ISSN 03772217
- [Ouyang und Madanat 2004] OUYANG, Yanfeng ; MADANAT, Samer: Optimal scheduling of rehabilitation activities for multiple pavement facilities: exact and approximate solutions. In: *Transportation Research Part A: Policy and Practice* 38 (2004), Nr. 5, 347–365. <http://dx.doi.org/10.1016/j.tra.2003.10.007>. – ISSN 09658564
- [Ouyang und Madanat 2006] OUYANG, Yanfeng ; MADANAT, Samer: An analytical solution for the finite-horizon pavement resurfacing planning problem. In: *Transportation Research Part B: Methodological* 40 (2006), Nr. 9, 767–778. <http://dx.doi.org/10.1016/j.trb.2005.11.001>. – ISSN 01912615
- [Pantelias und Zhang 2009] PANTELIAS, Aristeidis ; ZHANG, Zhanmin: Quadratic Linear Control Approach to Managing Network-Level Performance Goals of Pavement Infrastructure Systems. In: *Transportation Research Record: Journal of the Transportation Research Board* 2093 (2009), Nr. 1, 57–66. <http://dx.doi.org/10.3141/2093-07>. – ISSN 0361–1981
- [Patidar u. a. 2011] PATIDAR, Vandana ; LABI, Samuel ; MORIN, Thomas ; THOMPSON, Paul D. ; SINHA, Kumares C.: Evaluating Methods and Algorithms for Multicriteria Bridge Management at the Network Level. In: *Transportation Research Record: Journal of the Transportation Research Board* 2220 (2011), Nr. 1, 38–47. <http://dx.doi.org/10.3141/2220-05>. – ISSN 0361–1981
- [Petchrompo und Parlikad 2019] PETCHROMPO, Sanyapong ; PARLIKAD, Ajith K.: A review of asset management literature on multi-asset systems. In: *Reliability Engineering & System Safety* 181 (2019), 181–201. <http://dx.doi.org/10.1016/j.ress.2018.09.009>. – ISSN 09518320
- [Powell 2011] POWELL, Warren B.: *Approximate Dynamic Programming*. 2. Aufl. Hoboken, NJ, USA : John Wiley & Sons, Inc., 2011 (Wiley Series in Probability and Statistics). <http://dx.doi.org/10.1002/9781118029176>. – ISBN 9781118029176
- [Richter 2019] RICHTER, Stefan: *Statistisches und maschinelles Lernen*. Berlin, Heidelberg : Springer Berlin Heidelberg, 2019. <http://dx.doi.org/10.1007/978-3-662-59354-7>. – ISBN 978–3–662–59353–0
- [Roshandeh u. a. 2015] ROSHANDEH, Arash M. ; LI, Zongzhi ; NEISHAPOURI, Mohammad ; PATEL, Harshingar ; LIU, Yi: Trade-Off Analysis Approach for Multiobjective



- Transportation Investment Decision Making. In: *Journal of Transportation Engineering* 141 (2015), Nr. 3, 04014085. [http://dx.doi.org/10.1061/\(ASCE\)TE.1943-5436.0000747](http://dx.doi.org/10.1061/(ASCE)TE.1943-5436.0000747). – ISSN 0733-947X
- [Ruschel u. a. 2017] RUSCHEL, Edson ; SANTOS, Eduardo Alves P. ; LOURES, Eduardo de Freitas R.: Industrial maintenance decision-making: A systematic literature review. In: *Journal of Manufacturing Systems* 45 (2017), 180–194. <http://dx.doi.org/10.1016/j.jmsy.2017.09.003>. – ISSN 02786125
- [Russell und Norvig 2016] RUSSELL, Stuart J. ; NORVIG, Peter: *Artificial Intelligence A Modern Approach*. Pearson Education, 2016. – ISBN 9780136042594
- [Ryll und Freund 2010] RYLL, Frank ; FREUND, Curt: Grundlagen der Instandhaltung. 2010. In: *Instandhaltung technischer Systeme*. Berlin, Heidelberg : Springer Berlin Heidelberg, 2010. [http://dx.doi.org/10.1007/978-3-642-03949-2\\_2](http://dx.doi.org/10.1007/978-3-642-03949-2_2), 23–101
- [Sadek u. a. 2003] SADEK, Adel W. ; KVASNAK, Andrea ; SEGALE, Joe: Integrated Infrastructure Management Systems: Small Urban Area's Experience. In: *Journal of Infrastructure Systems* 9 (2003), Nr. 3, 98–106. [http://dx.doi.org/10.1061/\(ASCE\)1076-0342\(2003\)9:3\(98\)](http://dx.doi.org/10.1061/(ASCE)1076-0342(2003)9:3(98)). – ISSN 1076-0342
- [Sakib und Wuest 2018] SAKIB, Nazmus ; WUEST, Thorsten: Challenges and Opportunities of Condition-based Predictive Maintenance: A Review. In: *Procedia CIRP* 78 (2018), 267–272. <http://dx.doi.org/10.1016/j.procir.2018.08.318>. – ISBN 9781510875692
- [Santosa und Symes 1986] SANTOSA, Fadil ; SYMES, William W.: Linear Inversion of Band-Limited Reflection Seismograms. In: *SIAM Journal on Scientific and Statistical Computing* 7 (1986), Nr. 4, 1307–1330. <http://dx.doi.org/10.1137/0907087>. – ISSN 0196-5204
- [Sarkale u. a. 2018] SARKALE, Yugandhar ; NOZHATI, Saeed ; P. CHONG, Edwin K. ; ELLINGWOOD, Bruce R. ; MAHMOUD, Hussam: Solving Markov decision processes for network-level post-hazard recovery via simulation optimization and rollout. In: *2018 IEEE 14th International Conference on Automation Science and Engineering (CASE)* Bd. 2018-Augus, IEEE, 2018. – ISBN 978-1-5386-3593-3, 906–912
- [Sathaye und Madanat 2011] SATHAYE, Nakul ; MADANAT, Samer: A bottom-up solution for the multi-facility optimal pavement resurfacing problem. In: *Transportation Research*

- Part B: Methodological* 45 (2011), Nr. 7, 1004–1017. <http://dx.doi.org/10.1016/j.trb.2011.03.002>. – ISSN 01912615
- [Sathaye und Madanat 2012] SATHAYE, Nakul ; MADANAT, Samer: A bottom-up optimal pavement resurfacing solution approach for large-scale networks. In: *Transportation Research Part B: Methodological* 46 (2012), Nr. 4, 520–528. <http://dx.doi.org/10.1016/j.trb.2011.12.001>. – ISSN 01912615
- [Scholl 2001] SCHOLL, Armin: *Robuste Planung und Optimierung*. Berlin Heidelberg : Springer-Verlag, 2001. – 404 S. <http://dx.doi.org/10.1007/978-3-642-57570-9>. – ISBN 978-3-642-63303-4
- [Šelih u. a. 2008] ŠELIH, Jana ; KNE, Anžej ; SRDIĆ, Aleksander ; ŽURA, Marjan: Multiple-Criteria Decision Support System In Highway Infrastructure Management. In: *TRANSPORT* 23 (2008), Nr. 4, 299–305. <http://dx.doi.org/10.3846/1648-4142.2008.23.299-305>. – ISSN 1648-4142
- [Shi u. a. 2021] SHI, Yue ; XIANG, Yisha ; XIAO, Hui ; XING, Liudong: Joint optimization of budget allocation and maintenance planning of multi-facility transportation infrastructure systems. In: *European Journal of Operational Research* 288 (2021), Nr. 2, 382–393. <http://dx.doi.org/10.1016/j.ejor.2020.05.050>. – ISSN 03772217
- [Silver u. a. 2014] SILVER, David ; LEVER, Guy ; HEESS, Nicolas ; DEGRIS, Thomas ; WIERSTRA, Daan ; RIEDMILLER, Martin: Deterministic policy gradient algorithms. In: *31st International Conference on Machine Learning, ICML 2014* 1 (2014), S. 605–619. ISBN 9781634393973
- [van Suntum 2005] SUNTUM, Ulrich van: *Die unsichtbare Hand*. Berlin, Heidelberg : Springer Berlin Heidelberg, 2005. <http://dx.doi.org/10.1007/3-540-27688-2>. – ISBN 978-3-540-25235-1
- [Sutton und Barto 2018] SUTTON, Richard S. ; BARTO, Andrew G.: *Reinforcement Learning*. 2018. – 481 S. – ISBN 9780262039246
- [Swei u. a. 2019] SWEI, Omar ; GREGORY, Jeremy ; KIRCHAIN, Randolph: Embedding Flexibility within Pavement Management: Technique to Improve Expected Performance of Roadway Systems. In: *Journal of Infrastructure Systems* 25 (2019), Nr. 3, 05019007. [http://dx.doi.org/10.1061/\(ASCE\)IS.1943-555X.0000496](http://dx.doi.org/10.1061/(ASCE)IS.1943-555X.0000496). – ISSN 1076-0342
- [Thomas 1986] THOMAS, L.C.: A survey of maintenance and replacement models for maintainability and reliability of multi-item systems. In: *Reliability Engineering*



- 16 (1986), Nr. 4, 297–309. [http://dx.doi.org/10.1016/0143-8174\(86\)90099-5](http://dx.doi.org/10.1016/0143-8174(86)90099-5). – ISSN 01438174
- [Tibshirani 1996] TIBSHIRANI, Robert: Regression Shrinkage and Selection Via the Lasso. In: *Journal of the Royal Statistical Society: Series B (Methodological)* 58 (1996), Nr. 1, 267–288. <http://dx.doi.org/10.1111/j.2517-6161.1996.tb02080.x>. – ISSN 00359246
- [Tibshirani u. a. 2004] TIBSHIRANI, Robert ; JOHNSTONE, Iain ; HASTIE, Trevor ; EFRON, Bradley: Least angle regression. In: *The Annals of Statistics* 32 (2004), Nr. 2, 407–499. <http://dx.doi.org/10.1214/009053604000000067>. – ISSN 0090–5364
- [Torres-Machí u. a. 2014] TORRES-MACHÍ, Cristina ; CHAMORRO, Alondra ; VIDELA, Carlos ; PELLICER, Eugenio ; YEPES, Víctor: An Iterative Approach for the Optimization of Pavement Maintenance Management at the Network Level. In: *The Scientific World Journal* 2014 (2014), 1–11. <http://dx.doi.org/10.1155/2014/524329>. – ISSN 2356–6140
- [Truong u. a. 2018] TRUONG, Tung ; LI, Zongzhi ; KEPAPTSOGLOU, Konstantinos: Entropy-STEP Multiobjective Trade-Off Analysis Method for Optimal Transportation Investment Decisions. In: *Journal of Transportation Engineering, Part A: Systems* 144 (2018), Nr. 1, 04017065. <http://dx.doi.org/10.1061/JTEPBS.0000100>. – ISSN 2473–2907
- [Volle 1997] VOLLE, Michel: Quasiconvex Duality for the Max of Two Functions. 1997. 1997. [http://dx.doi.org/10.1007/978-3-642-59073-3\\_26](http://dx.doi.org/10.1007/978-3-642-59073-3_26), 365–379
- [van der Waerden 1993] WAERDEN, B. L. d.: *Algebra I*. Berlin, Heidelberg : Springer Berlin Heidelberg, 1993. <http://dx.doi.org/10.1007/978-3-642-85527-6>. – ISBN 978–3–642–85528–3
- [Wang u. a. 2003] WANG, Feng ; ZHANG, Zhanmin ; MACHEMEHL, Randy B.: Decision-Making Problem for Managing Pavement Maintenance and Rehabilitation Projects. In: *Transportation Research Record: Journal of the Transportation Research Board* 1853 (2003), Nr. 1, 21–28. <http://dx.doi.org/10.3141/1853-03>. – ISSN 0361–1981
- [Wang 2002] WANG, Hongzhou: A survey of maintenance policies of deteriorating systems. In: *European Journal of Operational Research* 139 (2002), Nr. 3, 469–489. [http://dx.doi.org/10.1016/S0377-2217\(01\)00197-7](http://dx.doi.org/10.1016/S0377-2217(01)00197-7). – ISSN 03772217

- [Wang u. a. 2007] WANG, Kelvin C. P. ; NGUYEN, Vu ; ZANIEWSKI, John P.: Genetic Algorithms-Based Network Optimization System with Multiple Objectives. In: *Transportation Research Record: Journal of the Transportation Research Board* 2016 (2007), Nr. 1, 85–95. <http://dx.doi.org/10.3141/2016-10>. – ISBN 9780309104432
- [Wang und Chen 2016] WANG, Rui ; CHEN, Nan: A survey of condition-based maintenance modeling of multi-component systems. In: *2016 IEEE International Conference on Industrial Engineering and Engineering Management (IEEM)* Bd. 2016-Decem, IEEE, 2016. – ISBN 978-1-5090-3665-3, 1664–1668
- [Watkins und Dayan 1992] WATKINS, Christopher J. C. H. ; DAYAN, Peter: Q-learning. In: *Machine Learning* 8 (1992), Nr. 3-4, 279–292. <http://dx.doi.org/10.1007/BF00992698>. – ISSN 0885-6125
- [Webel und Wied 2016] WEBEL, Karsten ; WIED, Dominik: *Stochastische Prozesse*. Wiesbaden : Springer Fachmedien Wiesbaden, 2016. <http://dx.doi.org/10.1007/978-3-658-13885-1>. – ISBN 978-3-658-13884-4
- [Wilcoxon 1945] WILCOXON, Frank: Individual Comparisons by Ranking Methods. In: *Biometrics Bulletin* 1 (1945), Nr. 6, S. 80. <http://dx.doi.org/10.2307/3001968>. – ISBN 0006341X
- [Wu u. a. 2012] WU, Zheng ; FLINTSCH, Gerardo ; FERREIRA, Adelino ; PICADO-SANTOS, Luís de: Framework for Multiobjective Optimization of Physical Highway Assets Investments. In: *Journal of Transportation Engineering* 138 (2012), dec, Nr. 12, 1411–1421. [http://dx.doi.org/10.1061/\(ASCE\)TE.1943-5436.0000458](http://dx.doi.org/10.1061/(ASCE)TE.1943-5436.0000458). – ISSN 0733-947X
- [Yadollahi u. a. 2015] YADOLLAHI, Mohammadreza ; ABD MAJID, Muhd Z. ; MOHAMAD ZIN, Rosli: Post-Pareto optimality approach to enhance budget allocation process for bridge rehabilitation management. In: *Structure and Infrastructure Engineering* 11 (2015), Nr. 12, 1565–1582. <http://dx.doi.org/10.1080/15732479.2014.980833>. – ISSN 1573-2479
- [Zhang u. a. 2018] ZHANG, Chiyuan ; VINYALS, Oriol ; MUNOS, Remi ; BENGIO, Samy: A Study on Overfitting in Deep Reinforcement Learning. In: *arXiv* (2018), 1–25. <http://arxiv.org/abs/1804.06893>. – ISSN 23318422

# A

## Mathematischer Anhang

### A.1 Verschleiß einfacher binärer Systeme

*Zuverlässigkeit* fassen wir als „Intaktwahrscheinlichkeit“ auf, konkret als die Wahrscheinlichkeit, dass eine Komponente oder ein System (vgl. Abschnitt 2.3) zu einem bestimmten Zeitpunkt noch nicht defekt ist. *Verfügbarkeit* hingegen ist die Wahrscheinlichkeit, dass eine Komponente oder ein System zu einem bestimmten Zeitpunkt betriebsbereit (und damit nicht in Reparatur) ist.

Die *Lebensdauer*  $X$  eines einfachen binären Systems ist nach Beichelt (1993, S. 15) die Zeitspanne von der Inbetriebnahme des Systems bis zum ersten Ausfall. Für den Systemzustand  $z(t) \in \{0; 1\}$  zum Zeitpunkt  $t$  gilt also

$$z(t) = \begin{cases} 1 & \text{für } t < X, \\ 0 & \text{für } t \geq X. \end{cases}$$

Die *Ausfallwahrscheinlichkeit*  $F(t)$  eines einfachen binären Systems bis zu einem Zeitpunkt  $t$  ist somit (vgl. Beichelt (1993, S. 16)) die Verteilungsfunktion der Lebensdauer

$$F(t) = \mathbb{P}(X < t)$$

mit zugehöriger Dichte

$$f(t) = \frac{dF(t)}{dt},$$

wenn  $F$  differenzierbar ist.

Die *Zuverlässigkeitsfunktion* (bzw. *Überlebenswahrscheinlichkeit*)  $\bar{F}(t)$  (vgl. Beichelt (1993, S. 17)) ist somit die Gegenwahrscheinlichkeit zur Ausfallwahrscheinlichkeit und definiert als

$$\bar{F}(t) = \mathbb{P}(X \geq t) = 1 - F(t).$$

Damit haben wir können wir nach Beichelt (1993, S. 18) den Zusammenhang zur erwarteten Lebensdauer  $\mathbb{E}(X)$  folgendermaßen beschreiben:

$$\mathbb{E}(X) = \int_0^\infty R(x) dx$$

Die *Ausfallrate*  $r(t)$ , die die Ausfallwahrscheinlichkeit zu einem Zeitpunkt  $t$  beschreibt (vgl. Barlow und Proschan (1966, S. 10) bzw. Beichelt (1993, S. 22)) ist definiert als

$$r(t) = \frac{f(t)}{\bar{F}(t)}.$$

Die Lebensdauer bei einfachen binären Systemen wird häufig annähernd vereinfacht als exponential- (ohne Alterung, vgl. Nelson (1982, S. 18 ff.) oder Beichelt (1993, S. 20 f.)) oder als Weibull-verteilt (mit Alterung, vgl. Nelson (1982, S. 36 ff.) oder Beichelt (1993, S. 20 f.)) angenommen. Weitere sinnvolle Verteilungen sind nach Nelson (1982, S. 28ff.) vor allem die Normal- und die Lognormal- oder auch die Rayleigh-Verteilung (vgl. Beichelt (1993, S. 17 ff.)).

Der Verschleiß zu einem Zeitpunkt  $t$  kann also mit dem jeweiligen Wert der Funktion  $r(t)$  assoziiert werden. Über den Zeitverlauf erfolgt dies bei einfachen binären Systemen nach Beichelt (1993, S. 25) häufig in der Form einer Badewannenkurve (vgl. Abb. A.1), die so aufgrund des üblichen Kurvenverlaufs von  $r(t)$  bei einer Weibull-Verteilung genannt wird.

Dann ist

$$r(t) = \lambda \beta(t) t^{\beta(t)-1}, \tag{A.1}$$

wobei gilt, dass in der ersten Phase (1)  $\beta(t) < 1$ , wodurch die Ausfallrate sinkt und damit die Frühfehler beschreibt, in einer zweiten Phase (2) gilt annähernd  $\beta(t) \approx 1$  und damit

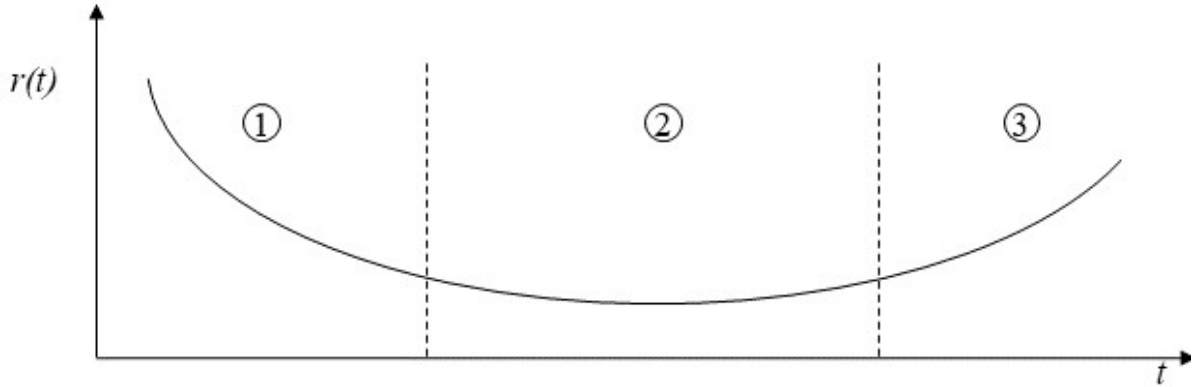


Abb. A.1: Verlauf der Ausfallrate als Badewannenkurve

Quelle: Eigene Darstellung in Anlehnung an Beichelt (1993, S. 25)

annähernd  $r(t) \approx \lambda$ , in der letzten Phase (3) ist  $\beta(t) > 1$ , wodurch die Ausfallrate steigt und Alterungseffekte beschreibt. Phase (2) mit  $\beta(t) = \beta = 1$  und  $r(t) \equiv \lambda$  entspricht dem Ausfallverhalten der Exponentialverteilung.

## A.2 Generalisierbarkeit der Investitionsfunktion

An dieser Stelle möchten wir noch einmal verifizieren, dass die Herleitung der Investitionsfunktion für alle  $0 < x' \leq x$  sinnvoll ist und nicht nur für  $x' \downarrow 0$ :

$$\begin{aligned}
 g^+(x, p) &\stackrel{!}{=} g^+(x - x', g^+(x', p)) \\
 1 - (1 - p) \cdot \exp(-\hat{a}x) &= 1 - (1 - (1 - (1 - p) \cdot \exp(-\hat{a}x'))) \cdot \exp(-\hat{a} \cdot (x - x')) \\
 (1 - p) \cdot \exp(-\hat{a}x) &= (1 - (1 - (1 - p) \cdot \exp(-\hat{a}x'))) \cdot \exp(-\hat{a} \cdot (x - x')) \\
 (1 - p) &= (1 - (1 - (1 - p) \cdot \exp(-\hat{a}x'))) \cdot \exp(\hat{a}x') \\
 (1 - p) &= \exp(\hat{a}x') - \exp(\hat{a}x') \cdot (1 - (1 - p) \cdot \exp(-\hat{a}x')) \\
 (1 - p) &= \exp(\hat{a}x') - \exp(\hat{a}x') + (1 - p) \cdot \exp(\hat{a}x') \cdot \exp(-\hat{a}x') \\
 (1 - p) &= \exp(\hat{a}x') - \exp(\hat{a}x') + (1 - p) \cdot \exp(\hat{a}x' - \hat{a}x) \\
 (1 - p) &= \exp(\hat{a}x') - \exp(\hat{a}x') + (1 - p) \cdot \exp(\hat{a}x' - \hat{a}x)
 \end{aligned}$$

## A.3 Optimalitätsbedingungen

### A.3.1 Zwei-Komponenten-Parallelsystem, eine Periode

Analyse der zweiten Ableitung für die Zielfunktion des Parallelsystems

Berechne die zweite Ableitung

$$\begin{aligned}
 & \frac{\partial^2}{\partial x_1^2} \left( (1 - \hat{p}_1) \cdot (1 - \hat{p}_2) \cdot \exp \left( -\hat{a}_1 x_1 - \hat{a}_2 \hat{B} + \hat{a}_2 x_1 \right) \right. \\
 & \quad \left. + \widehat{\Delta w}_1 \cdot (1 - \hat{p}_2) \cdot \exp \left( -\hat{a}_2 \hat{B} + \hat{a}_2 x_1 \right) + \widehat{\Delta w}_2 \cdot (1 - \hat{p}_1) \cdot \exp \left( -\hat{a}_1 x_1 \right) \right) \\
 &= \frac{\partial}{\partial x_1} \left( (1 - \hat{p}_1) \cdot (1 - \hat{p}_2) \cdot \exp \left( -\hat{a}_2 \hat{B} \right) \cdot (\hat{a}_2 - \hat{a}_1) \cdot \exp \left( (\hat{a}_2 - \hat{a}_1) x_1 \right) \right. \\
 & \quad \left. + \widehat{\Delta w}_1 \cdot (1 - \hat{p}_2) \cdot \exp \left( -\hat{a}_2 \hat{B} \right) \cdot \hat{a}_2 \cdot \exp \left( \hat{a}_2 x_1 \right) \right. \\
 & \quad \left. - \widehat{\Delta w}_2 \cdot (1 - \hat{p}_1) \cdot \hat{a}_1 \cdot \exp \left( -\hat{a}_1 x_1 \right) \right) \\
 &= (1 - \hat{p}_1) \cdot (1 - \hat{p}_2) \cdot \exp \left( -\hat{a}_2 \hat{B} \right) \cdot (\hat{a}_2 - \hat{a}_1)^2 \cdot \exp \left( (\hat{a}_2 - \hat{a}_1) x_1 \right) \\
 & \quad + \widehat{\Delta w}_1 \cdot (1 - \hat{p}_2) \cdot \exp \left( -\hat{a}_2 \hat{B} \right) \cdot (\hat{a}_2)^2 \cdot \exp \left( \hat{a}_2 x_1 \right) \\
 & \quad + \widehat{\Delta w}_2 \cdot (1 - \hat{p}_1) \cdot (\hat{a}_1)^2 \cdot \exp \left( -\hat{a}_1 x_1 \right) \geq 0
 \end{aligned}$$

### A.3.2 Zwei-Komponenten-Seriensystem, eine Periode

Wir setzen wieder  $x_2 = \hat{B} - x_1$

$$\begin{aligned}
 & \max f(x_1) \\
 &= \left( 1 - (1 - \hat{p}_1) \cdot \exp \left( -\hat{a}_1 x_1 \right) - \widehat{\Delta w}_1 \right) \cdot \left( 1 - (1 - \hat{p}_2) \cdot \exp \left( -\hat{a}_2 \cdot (\hat{B} - x_1) \right) \right) - \widehat{\Delta w}_2
 \end{aligned}$$

Berechne Ableitung  $f'(x)$  und setze diese gleich 0, um Kandidaten für lokale Optima zu bestimmen.

$$\frac{\partial}{\partial x_1} f(x_1)$$

$$\begin{aligned}
&= \frac{\partial}{\partial x_1} \left( 1 - (1 - \hat{p}_2) \cdot \exp \left( -\hat{a}_2 \cdot (\hat{B} - x_1) \right) \right) - \widehat{\Delta w}_2 - (1 - \hat{p}_1) \cdot \exp \left( -\hat{a}_1 x_1 \right) \\
&\quad + (1 - \hat{p}_1) \cdot (1 - \hat{p}_2) \cdot \exp \left( -\hat{a}_1 x_1 - \hat{a}_2 \hat{B} + \hat{a}_2 x_1 \right) \\
&\quad + \widehat{\Delta w}_2 \cdot (1 - \hat{p}_1) \cdot \exp \left( -\hat{a}_1 x_1 \right) - \widehat{\Delta w}_1 \\
&\quad + \widehat{\Delta w}_1 \cdot (1 - \hat{p}_2) \cdot \exp \left( -\hat{a}_2 \hat{B} + \hat{a}_2 x_1 \right) + \widehat{\Delta w}_1 \widehat{\Delta w}_2
\end{aligned}$$

Da die Konstanten bei der Maximierung keinen Einfluss haben, kann die Formel vereinfacht werden zu

$$\begin{aligned}
&\frac{\partial}{\partial x_1} \bar{f}(x_1) \\
&= \frac{\partial}{\partial x_1} \left( - (1 - \hat{p}_2) \cdot \exp \left( -\hat{a}_2 \cdot (\hat{B} - x_1) \right) - (1 - \hat{p}_1) \cdot \exp \left( -\hat{a}_1 x_1 \right) \right. \\
&\quad \left. + (1 - \hat{p}_1) \cdot (1 - \hat{p}_2) \cdot \exp \left( -\hat{a}_1 x_1 - \hat{a}_2 \hat{B} + \hat{a}_2 x_1 \right) \right. \\
&\quad \left. + \widehat{\Delta w}_2 \cdot (1 - \hat{p}_1) \cdot \exp \left( -\hat{a}_1 x_1 \right) + \widehat{\Delta w}_1 \cdot (1 - \hat{p}_2) \cdot \exp \left( -\hat{a}_2 \hat{B} + \hat{a}_2 x_1 \right) \right)
\end{aligned}$$

mit Ableitung

$$\begin{aligned}
&= - (1 - \hat{p}_2) \cdot \exp \left( -\hat{a}_2 \hat{B} \right) \cdot \hat{a}_2 \cdot \exp \left( \hat{a}_2 x_1 \right) \\
&\quad + (1 - \hat{p}_1) \cdot \hat{a}_1 \cdot \exp \left( -\hat{a}_1 x_1 \right) \\
&\quad + (1 - \hat{p}_1) \cdot (1 - \hat{p}_2) \cdot \exp \left( -\hat{a}_2 \hat{B} \right) \cdot (\hat{a}_2 - \hat{a}_1) \cdot \exp \left( (\hat{a}_2 - \hat{a}_1) x_1 \right) \\
&\quad - \widehat{\Delta w}_2 \cdot (1 - \hat{p}_1) \cdot \hat{a}_1 \cdot \exp \left( -\hat{a}_1 x_1 \right) \\
&\quad + \widehat{\Delta w}_1 \cdot (1 - \hat{p}_2) \cdot \exp \left( -\hat{a}_2 \hat{B} \right) \cdot \hat{a}_2 \cdot \exp \left( \hat{a}_2 x_1 \right) \stackrel{!}{=} 0
\end{aligned}$$

## Analyse der Optimalitätsbedingung 2. Ordnung

Berechne die zweite Ableitung

$$\begin{aligned}
&\frac{\partial^2}{\partial x_1^2} \bar{f}(x_1) \\
&= \frac{\partial^2}{\partial x_1^2} \left( - (1 - \hat{p}_2) \cdot \exp \left( -\hat{a}_2 \cdot (\hat{B} - x_1) \right) - (1 - \hat{p}_1) \cdot \exp \left( -\hat{a}_1 x_1 \right) \right)
\end{aligned}$$

$$\begin{aligned}
& + (1 - \hat{p}_1) \cdot (1 - \hat{p}_2) \cdot \exp \left( -\hat{a}_1 x_1 - \hat{a}_2 \hat{B} + \hat{a}_2 x_1 \right) \\
& + \widehat{\Delta w}_2 \cdot (1 - \hat{p}_1) \cdot \exp \left( -\hat{a}_1 x_1 \right) + \widehat{\Delta w}_1 \cdot (1 - \hat{p}_2) \cdot \exp \left( -\hat{a}_2 \hat{B} + \hat{a}_2 x_1 \right) \\
& = \frac{\partial}{\partial x_1} \left( - \left( 1 - \widehat{\Delta w}_1 \right) (1 - \hat{p}_2) \cdot \exp \left( -\hat{a}_2 \hat{B} \right) \cdot \hat{a}_2 \cdot \exp \left( \hat{a}_2 x_1 \right) \right. \\
& \quad + \left( 1 - \widehat{\Delta w}_2 \right) \cdot (1 - \hat{p}_1) \cdot \hat{a}_1 \cdot \exp \left( -\hat{a}_1 x_1 \right) \\
& \quad + (1 - \hat{p}_1) \cdot (1 - \hat{p}_2) \cdot \exp \left( -\hat{a}_2 \hat{B} \right) \cdot (\hat{a}_2 - \hat{a}_1) \cdot \exp \left( (\hat{a}_2 - \hat{a}_1) x_1 \right) \\
& = - \left( 1 - \widehat{\Delta w}_1 \right) (1 - \hat{p}_2) \cdot \exp \left( -\hat{a}_2 \hat{B} \right) \cdot (\hat{a}_2)^2 \cdot \exp \left( \hat{a}_2 x_1 \right) \\
& \quad - \left( 1 - \widehat{\Delta w}_2 \right) \cdot (1 - \hat{p}_1) \cdot (\hat{a}_1)^2 \cdot \exp \left( -\hat{a}_1 x_1 \right) \\
& \quad + (1 - \hat{p}_1) \cdot (1 - \hat{p}_2) \cdot \exp \left( -\hat{a}_2 \hat{B} \right) \cdot (\hat{a}_2 - \hat{a}_1)^2 \cdot \exp \left( (\hat{a}_2 - \hat{a}_1) x_1 \right)
\end{aligned}$$

$$\begin{aligned}
& \Rightarrow \frac{\partial^2}{\partial x_1^2} \bar{f}(x_1) \leq 0 \\
& \Leftrightarrow (1 - \hat{p}_1) \cdot (1 - \hat{p}_2) \cdot \exp \left( -\hat{a}_2 \hat{B} \right) \cdot (\hat{a}_2 - \hat{a}_1)^2 \cdot \exp \left( (\hat{a}_2 - \hat{a}_1) x_1 \right) \\
& \leq \left( 1 - \widehat{\Delta w}_1 \right) (1 - \hat{p}_2) \cdot \exp \left( -\hat{a}_2 \hat{B} \right) \cdot (\hat{a}_2)^2 \cdot \exp \left( \hat{a}_2 x_1 \right) \\
& \quad + \left( 1 - \widehat{\Delta w}_2 \right) \cdot (1 - \hat{p}_1) \cdot (\hat{a}_1)^2 \cdot \exp \left( -\hat{a}_1 x_1 \right) \\
& \Leftrightarrow (1 - \hat{p}_1) \cdot (1 - \hat{p}_2) \cdot \exp \left( -\hat{a}_2 \hat{B} \right) \cdot (\hat{a}_2 - \hat{a}_1)^2 \cdot y^{\hat{a}_2 - \hat{a}_1} \\
& \leq \left( 1 - \widehat{\Delta w}_1 \right) (1 - \hat{p}_2) \cdot \exp \left( -\hat{a}_2 \hat{B} \right) \cdot (\hat{a}_2)^2 \cdot y^{\hat{a}_2} \\
& \quad + \left( 1 - \widehat{\Delta w}_2 \right) \cdot (1 - \hat{p}_1) \cdot (\hat{a}_1)^2 \cdot y^{-\hat{a}_1}
\end{aligned}$$

### Analytische Berechnung der Lösung

Wir transformieren die Optimierungsvariable, indem wir  $y := \exp(x_1)$ , und sortieren etwas um, wodurch

$$\begin{aligned}
& (1 - \hat{p}_1) \cdot \hat{a}_1 \cdot y^{-\hat{a}_1} \\
& + (1 - \hat{p}_1) \cdot (1 - \hat{p}_2) \cdot \exp \left( -\hat{a}_2 \hat{B} \right) \cdot (\hat{a}_2 - \hat{a}_1) \cdot y^{\hat{a}_2 - \hat{a}_1} \\
& + \widehat{\Delta w}_1 \cdot (1 - \hat{p}_2) \cdot \exp \left( -\hat{a}_2 \hat{B} \right) \cdot \hat{a}_2 \cdot y^{\hat{a}_2} \\
& = (1 - \hat{p}_2) \cdot \exp \left( -\hat{a}_2 \hat{B} \right) \cdot \hat{a}_2 \cdot y^{\hat{a}_2}
\end{aligned}$$



$$+ \widehat{\Delta w}_2 \cdot (1 - \hat{p}_1) \cdot \hat{a}_1 \cdot y^{-\hat{a}_1}$$

Wir multiplizieren die Gleichung mit  $y^{\hat{a}_1}$  und setzen  $\hat{a}_1 := m \cdot \hat{a}_2$  mit  $m > 0$ , da  $\hat{a}_1, \hat{a}_2, y^{\hat{a}_1} > 0$ . Dies führt uns zur Gleichung

$$\begin{aligned} & (1 - \hat{p}_1) \cdot m \cdot \hat{a}_2 \\ & + (1 - \hat{p}_1) \cdot (1 - \hat{p}_2) \cdot \exp(-\hat{a}_2 \hat{B}) \cdot (\hat{a}_2 - m \cdot \hat{a}_2) \cdot y^{\hat{a}_2} \\ & + \widehat{\Delta w}_1 \cdot (1 - \hat{p}_2) \cdot \exp(-\hat{a}_2 \hat{B}) \cdot \hat{a}_2 \cdot y^{\hat{a}_2 \cdot (1+m)} \\ & = (1 - \hat{p}_2) \cdot \exp(-\hat{a}_2 \hat{B}) \cdot \hat{a}_2 \cdot y^{\hat{a}_2 \cdot (1+m)} \\ & + \widehat{\Delta w}_2 \cdot (1 - \hat{p}_1) \cdot m \cdot \hat{a}_2, \end{aligned}$$

die wir durch  $z := y^{\hat{a}_2}$  in ein Polynom vom Grad  $(1 + m)$  überführen:

$$\begin{aligned} & \underbrace{(\widehat{\Delta w}_1 - 1) \cdot (1 - \hat{p}_2) \cdot \exp(-\hat{a}_2 \hat{B}) \cdot \hat{a}_2 \cdot z^{1+m}}_{=: c_{1+m}} \\ & + \underbrace{(1 - \hat{p}_1) \cdot (1 - \hat{p}_2) \cdot \exp(-\hat{a}_2 \hat{B}) \cdot ((1 - m) \cdot \hat{a}_2) \cdot z}_{=: c_1(m)} \\ & + \underbrace{(1 - \widehat{\Delta w}_2) \cdot (1 - \hat{p}_1) \cdot m \cdot \hat{a}_2}_{=: c_0(m)} \\ & = 0 \end{aligned}$$

### Analyse für ausgewählte Verhältnisse der Skalierungsparameter der Investitionsfunktion

Wir analysieren das Polynom nur für verschiedene Werte von  $m$ .

$m = 1$  Die Diskriminante für quadratische Polynome berechnet sich mit Koeffizienten der Terme  $i$ -ter Ordnung  $c_i$  durch

$$\begin{aligned} D_2 &= c_1^2 - 4c_2c_0 = -4c_2c_0 \\ &= 4 \cdot (1 - \widehat{\Delta w_1}) \cdot (1 - \widehat{p}_2) \cdot \exp(-\widehat{a}_2\widehat{B}) \cdot \widehat{a}_2 \cdot (1 - \widehat{\Delta w_2}) (1 - \widehat{p}_1) \cdot \widehat{a}_2 > 0 \end{aligned}$$

Wegen  $c_1 = 0$  für  $m = 1$  ist eine der beiden Lösungen negativ und kommt daher nicht weiter in Frage:

$$z_{1,2} = \frac{-c_1 \pm \sqrt{D_2}}{2c_2} = \pm \frac{\sqrt{D_2}}{2c_2} \quad (\text{A.2})$$

$m = 2$  Die Diskriminante für kubische Polynome berechnet sich mit Koeffizienten der Terme  $i$ -ter Ordnung  $c_i$  durch

$$\begin{aligned} D_3 &= c_2^2c_1^2 - 4c_3c_1^3 - 4c_2^3c_0 + 18c_3c_2c_1c_0 - 27c_3^2c_0^2 \\ &= -4c_3c_1^3 - 27c_3^2c_0^2 < 0, \end{aligned}$$

denn  $c_3 = -(1 - \widehat{\Delta w_1}) \cdot (1 - \widehat{p}_2) \cdot \exp(-\widehat{a}_2\widehat{B}) \cdot \widehat{a}_2 < 0$ ,  $c_1 = -(1 - \widehat{p}_1) \cdot (1 - \widehat{p}_2) \cdot \exp(-\widehat{a}_2\widehat{B}) \cdot \widehat{a}_2 < 0$ , d. h., mindestens eine der Nullstellen ist komplex.

Eine konkrete analytische Berechnung der Lösungen kann mit Hilfe der Cardanischen Formeln erfolgen (vgl. van der Waerden (1993, S. 191 ff.)).

$m = 3$  Die Diskriminante für quartische Polynome berechnet sich mit Koeffizienten der Terme  $i$ -ter Ordnung  $c_i$  durch (Terme mit  $c_i = 0$  wurden schon entfernt):

$$D_4 = 256c_4^3c_0^3 - 27c_4^2c_1^4 < 0,$$

denn  $c_4 = -(1 - \widehat{\Delta w_1}) \cdot (1 - \widehat{p}_2) \cdot \exp(-\widehat{a}_2\widehat{B}) \cdot \widehat{a}_2 < 0$ ,  $c_0 = (1 - \widehat{\Delta w_2}) (1 - \widehat{p}_1) \cdot 3 \cdot \widehat{a}_2 > 0$ , d. h., mindestens eine der Nullstellen ist komplex. Auch für diesen Fall ist eine analytische Lösung noch möglich (vgl. van der Waerden (1993, S. 191 ff.) oder Carpenter (1966)).

### A.3.3 KKT-Bedingungen des einperiodigen Problems:

#### Hinreichende Optimalitätsbedingungen zweiter Ordnung

Sei  $\mathbf{p}(\mathbf{x}) = \mathbf{g}(\mathbf{x}, \hat{\mathbf{p}}, \widehat{\Delta \mathbf{w}})$ . Betrachten wir nun zusätzlich die hinreichenden Bedingungen zweiter Ordnung (vgl. Bazaraa u. a. (2006, S. 211 ff.)) erneut für das *Minimierungsproblem* (5.24) auf S. 59. Für die zweiten partiellen Ableitungen des Grenznutzens haben wir mit Hilfe der Vertauschung von Differentiationsoperatoren durch den Satz von Schwarz (vgl. Forster (2017, S. 68 f.)):

$$\begin{aligned}
 \frac{\partial^2}{\partial x_i^2} \varphi(\mathbf{p}(\mathbf{x})) &= \underbrace{\frac{\partial^2}{\partial p_i^2} \varphi(\mathbf{p}(\mathbf{x}))}_{=0} \cdot \underbrace{\left( \frac{\partial}{\partial x_i} p_i(x_i) \right)^2}_{>0} + \frac{\partial}{\partial p_i} \varphi(\mathbf{p}(\mathbf{x})) \cdot \frac{\partial^2}{\partial x_i^2} p_i(x_i) \\
 &= \underbrace{\frac{\partial}{\partial p_i} \varphi(\mathbf{p}(\mathbf{x}))}_{>0} \cdot \underbrace{\frac{\partial^2}{\partial x_i^2} p_i(x_i)}_{<0} < 0 \\
 \frac{\partial^2}{\partial x_i \partial x_j} \varphi(\mathbf{p}(\mathbf{x})) &= \frac{\partial}{\partial x_j} \left( \frac{\partial}{\partial p_i} \varphi(p_1(x_1), \dots, p_n(x_n)) \cdot \frac{\partial}{\partial x_i} p_i(x_i) \right) \\
 &= \frac{\partial}{\partial x_j} \left( \frac{\partial}{\partial p_i} \varphi(p_1(x_1), \dots, p_n(x_n)) \right) \cdot \frac{\partial}{\partial x_i} p_i(x_i) \\
 &\quad + \underbrace{\frac{\partial}{\partial x_j} \left( \frac{\partial}{\partial x_i} p_i(x_i) \right)}_{=0} \cdot \frac{\partial}{\partial p_i} \varphi(\mathbf{p}(\mathbf{x})) \\
 &\stackrel{\text{Satz von Schwarz}}{=} \underbrace{\frac{\partial^2}{\partial p_i \partial p_j} \varphi(p_1(x_1), \dots, p_n(x_n))}_{\text{Vorzeichen abhängig von } \varphi} \cdot \underbrace{\frac{\partial}{\partial x_j} p_j(x_j)}_{>0} \cdot \underbrace{\frac{\partial}{\partial x_i} p_i(x_i)}_{>0}
 \end{aligned}$$

Wir suchen ein striktes lokales Minimum unter den Kandidaten, die den KKT-Bedingungen genügen. Dazu betrachte den Kegel  $K$  aller Vektoren  $\mathbf{d} \neq \mathbf{0}$ , für die gilt, dass

$$\nabla \left( \hat{B} - \sum_{i=1}^n x_i \right)^\top \cdot \mathbf{d} = - \begin{pmatrix} 1 \\ \vdots \\ 1 \end{pmatrix} \cdot (d_1, \dots, d_n) = 0,$$

also  $K = \{\mathbf{d} \neq \mathbf{0} \in \mathbb{R}^n : \sum_{i=1}^n d_i = 0\}$ . Für ein striktes lokales Minimum gilt somit

$$\mathbf{d}^\top \mathbf{H} \mathbf{d} > 0$$

Für unterschiedliche Strukturbewertungsfunktionen bedeutet das folgendes:

$$\begin{aligned} \varphi = \varphi^{\text{Ser}} = \prod_{i=1}^n p_i &\Rightarrow \frac{\partial^2}{\partial p_i \partial p_j} \varphi(\mathbf{p}(\mathbf{x})) = \frac{\prod_{k=1}^n p_k}{p_i \cdot p_j} > 0 \\ \varphi = \varphi^{\text{Par}} = 1 - \prod_{i=1}^n (1 - p_i) &\Rightarrow \frac{\partial^2}{\partial p_i \partial p_j} \varphi(\mathbf{p}(\mathbf{x})) = -\frac{\prod_{k=1}^n (1 - p_k)}{(1 - p_i) \cdot (1 - p_j)} < 0 \end{aligned}$$

Die Hessematrix der Lagrangefunktion von Seriensystemen hat also die Form

$$\mathbf{H} = \begin{pmatrix} h_{11} & -h_{12} & \cdots & -h_{1n} \\ -h_{21} & h_{22} & \ddots & \vdots \\ \vdots & \ddots & \ddots & -h_{n-1,n} \\ -h_{n1} & \cdots & -h_{n,n-1} & h_{nn} \end{pmatrix},$$

die von Parallelsystemen die Form

$$\mathbf{H} = \begin{pmatrix} h_{11} & \cdots & h_{1n} \\ \vdots & \ddots & \vdots \\ h_{n1} & \cdots & h_{nn} \end{pmatrix}$$

mit Einträgen  $h_{ij} > 0$  und  $h_{ij} = h_{ji}$  für  $i \neq j$ .

Für ein Seriensystem mit  $n = 2$  Komponenten haben wir also

$$\begin{aligned} (d_1, -d_1) \cdot \begin{pmatrix} h_{11} & -h_{12} \\ -h_{12} & h_{22} \end{pmatrix} \cdot \begin{pmatrix} d_1 \\ -d_1 \end{pmatrix} &= (d_1 h_{11} + d_1 h_{12}, -d_1 h_{12} - d_1 h_{22}) \cdot \begin{pmatrix} d_1 \\ -d_1 \end{pmatrix} \\ &= d_1^2 \cdot (h_{11} + h_{12}) + d_1^2 \cdot (h_{12} + h_{22}) \\ &= d_1^2 \cdot (h_{11} + 2h_{12} + h_{22}) > 0. \end{aligned}$$

Für ein Parallelsystem mit  $n = 2$  Komponenten haben wir

$$\begin{aligned}
 (d_1, -d_1) \cdot \begin{pmatrix} h_{11} & h_{12} \\ h_{12} & h_{22} \end{pmatrix} \cdot \begin{pmatrix} d_1 \\ -d_1 \end{pmatrix} &= (d_1 h_{11} - d_1 h_{12}, d_1 h_{12} - d_1 h_{22}) \cdot \begin{pmatrix} d_1 \\ -d_1 \end{pmatrix} \\
 &= d_1^2 \cdot (h_{11} - h_{12}) + d_1^2 \cdot (h_{22} - h_{12}) \\
 &= d_1^2 \cdot (h_{11} + h_{22} - 2h_{12}).
 \end{aligned}$$

Für die Brückenschaltung kann man keine an dieser Stelle keine eindeutige Aussage ableiten, denn die zweite Ableitung kann kleiner oder größer als 0 sein.

$$\begin{aligned}
 \varphi &= \varphi^{\text{Brücke}} \\
 \Rightarrow \frac{\partial}{\partial p_5} \varphi(\mathbf{p}(\mathbf{x})) &= (1 - (1 - p_1) \cdot (1 - p_2)) \cdot (1 - (1 - p_3) \cdot (1 - p_4)) \\
 &\quad + (1 - p_1 \cdot p_3) \cdot (1 - p_2 \cdot p_4) - 1 \\
 \Rightarrow \frac{\partial}{\partial p_1} \varphi(\mathbf{p}(\mathbf{x})) &= p_5 \cdot (1 - p_2) \cdot (1 - (1 - p_3) \cdot (1 - p_4)) + (1 - p_5) \cdot p_3 \cdot (1 - p_2 \cdot p_4) \\
 \Rightarrow \frac{\partial}{\partial p_2} \varphi(\mathbf{p}(\mathbf{x})) &= p_5 \cdot (1 - p_1) \cdot (1 - (1 - p_3) \cdot (1 - p_4)) + (1 - p_5) \cdot p_4 \cdot (1 - p_1 \cdot p_3) \\
 \Rightarrow \frac{\partial}{\partial p_3} \varphi(\mathbf{p}(\mathbf{x})) &= p_5 \cdot (1 - p_4) \cdot (1 - (1 - p_1) \cdot (1 - p_2)) + (1 - p_5) \cdot p_1 \cdot (1 - p_2 \cdot p_4) \\
 \Rightarrow \frac{\partial}{\partial p_4} \varphi(\mathbf{p}(\mathbf{x})) &= p_5 \cdot (1 - p_3) \cdot (1 - (1 - p_1) \cdot (1 - p_2)) + (1 - p_5) \cdot p_2 \cdot (1 - p_1 \cdot p_3) \\
 \Rightarrow \frac{\partial^2}{\partial p_1 \partial p_5} \varphi(\mathbf{p}(\mathbf{x})) &= p_2 \cdot \underbrace{\left( p_3 \cdot \underbrace{(p_4 - 1)}_{<0} + p_4 \cdot (p_3 - 1) \right)}_{<0} + \underbrace{(1 - p_3) \cdot p_4}_{>0}
 \end{aligned}$$

### A.3.4 Analyse der KKT-Bedingungen des mehrperiodigen Problems

Wir formulieren zunächst das ursprüngliche mehrperiodige Optimierungsproblem etwas um und verwenden dabei explizit die Erkenntnisse aus Satz 1 auf Seite 46 und Satz 3 auf

Seite 51. Es verbleibt folgendes Optimierungsproblem in Standardform:

$$\begin{aligned}
 & \text{Min.}_{\mathbf{x}} \quad z \\
 & \text{u. d. N.} \\
 & z - \phi_t \leq 0 & (t = 1, \dots, T) \\
 & \phi_{t+1} - \phi_t \leq 0 & (t = 1, \dots, T-1) \\
 & \phi_t - \varphi \left( \mathbf{g} \left( \mathbf{x}_t, \mathbf{p}_{t-1}, \widehat{\Delta \mathbf{w}}_t \right) \right) = 0 & (t = 1, \dots, T) \\
 & \sum_{t=1}^T \sum_{i=1}^n x_{it} - \widehat{B} = 0 \\
 & z \geq 0 \\
 & x_{it} \geq 0 & (i = 1, \dots, n; \\
 & & t = 1, \dots, T) \\
 & \phi_t \geq 0 & (t = 1, \dots, T)
 \end{aligned}$$

Damit haben wir die Lagrangefunktion

$$\begin{aligned}
 L \left( \underbrace{(z, \mathbf{x}, \Phi)^{\top}}_{=: \mathbf{y} \in \mathbb{R}^{1+n \cdot T+T}}, \boldsymbol{\mu}, \boldsymbol{\lambda} \right) &= -z + \mu_T \cdot (z - \phi_T) + \sum_{t=1}^{T-1} \mu_t \cdot (\phi_{t+1} - \phi_t) \\
 &+ \sum_{t=1}^T \lambda_t^{\varphi} \cdot \left( \phi_t - \varphi \left( \mathbf{g} \left( \mathbf{x}_t, \mathbf{p}_{t-1}, \widehat{\Delta \mathbf{w}}_t \right) \right) \right) \\
 &+ \lambda^B \cdot \left( \sum_{t=1}^T \sum_{i=1}^n x_{it} - \widehat{B} \right)
 \end{aligned}$$

mit  $\Phi = (\phi_t)_{t=1, \dots, T}$  und die folgenden KKT-Bedingungen für das mehrperiodige Problem

$$\begin{aligned}
 & \nabla_{\mathbf{y}} L(\mathbf{y}, \boldsymbol{\mu}, \boldsymbol{\lambda}) \geq \mathbf{0} \\
 & z - \phi_T \leq 0 \\
 & \phi_{t+1} - \phi_t \leq 0 & (t = 1, \dots, T-1) \\
 & \phi_t - \varphi \left( \mathbf{g} \left( \mathbf{x}_t, \mathbf{p}_{t-1}, \widehat{\Delta \mathbf{w}}_t \right) \right) = 0 & (t = 1, \dots, T)
 \end{aligned}$$

$$\begin{aligned}
\sum_{t=1}^T \sum_{i=1}^n x_{it} - \widehat{B} &= 0 \\
\mu_t &\geq 0 & (t = 1, \dots, T) \\
\mu_t \cdot (z - \phi_t) &= 0 & (t = 1, \dots, T) \\
\mu_t \cdot (\phi_{t+1} - \phi_t) &= 0 & (t = 1, \dots, T-1) \\
y_l \cdot \nabla_{\mathbf{y}} L(\mathbf{y}, \boldsymbol{\mu}, \boldsymbol{\lambda}) &= 0 & (l = 1, \dots, (1 + n \cdot T + T)) \quad (\text{A.3}) \\
z &\geq 0 \\
x_{it} &\geq 0 & (i = 1, \dots, n; \\
& & t = 1, \dots, T) \\
\phi_t &\geq 0 & (t = 1, \dots, T)
\end{aligned}$$

Berechne nun die partiellen Ableitungen für den Gradienten der Lagrangefunktion:

$$\begin{aligned}
\frac{\partial}{\partial z} L(\mathbf{y}, \boldsymbol{\mu}, \boldsymbol{\lambda}) &= -1 + \mu_T \geq 0 \\
\frac{\partial}{\partial x_{it}} L(\mathbf{y}, \boldsymbol{\mu}, \boldsymbol{\lambda}) &= -\lambda_t^\varphi \cdot \frac{\partial}{\partial x_{it}} \left( \varphi \left( \mathbf{g} \left( \mathbf{x}_t, \mathbf{p}_{t-1}, \widehat{\Delta \mathbf{w}_t} \right) \right) \right) + \lambda^B \geq 0 & (i = 1, \dots, n; \\
& & t = 1, \dots, T) \\
\frac{\partial}{\partial \phi_1} L(\mathbf{y}, \boldsymbol{\mu}, \boldsymbol{\lambda}) &= -\mu_1 + \lambda_1^\varphi \geq 0 \\
\frac{\partial}{\partial \phi_t} L(\mathbf{y}, \boldsymbol{\mu}, \boldsymbol{\lambda}) &= \mu_{t-1} - \mu_t + \lambda_t^\varphi \geq 0 & (t = 2, \dots, T)
\end{aligned}$$

Wir nehmen  $\phi_t \geq 0$  ( $t = 1, \dots, T$ ) an und damit auch  $z \geq 0$ .

Aus den Komplementaritätsbedingungen (A.3) folgt zunächst

$$\mu_T = 1 \tag{A.4}$$

$$\lambda_1^\varphi = \mu_1$$

$$\lambda_t^\varphi = \mu_t - \mu_{t-1} \quad (t = 2, \dots, T) \tag{A.5}$$

Für  $x_{it} > 0$ , das den KKT-Bedingungen genügt, bedeutet das in einer Periode  $t \in$

$\{1, \dots, T\}$ , dass

$$\frac{\partial}{\partial x_{it}} \varphi \left( \mathbf{g} \left( \mathbf{x}_t, \mathbf{p}_{t-1}, \widehat{\Delta \mathbf{w}}_t \right) \right) = \frac{\lambda^B}{\lambda_t^\varphi} > 0 \quad \forall i = 1, \dots, n \quad (\text{A.6})$$

konstant ist. Für  $x_{it} = 0$  haben wir entsprechend, dass

$$\frac{\partial}{\partial x_{it}} \varphi \left( \mathbf{g} \left( \mathbf{x}_t, \mathbf{p}_{t-1}, \widehat{\Delta \mathbf{w}}_t \right) \right) \leq \frac{\lambda^B}{\lambda_t^\varphi}.$$

Aus der Gl. (A.6) kann man zudem für Periode  $t = 1$  aus  $\frac{\lambda^B}{\lambda_1^\varphi} = \frac{\lambda^B}{\mu_1} > 0$  und  $\mu_1 \geq 0$  schließen, dass  $\lambda^B > 0$ . Somit sind auch wegen Gl. (A.6) alle weiteren  $\lambda_t^\varphi > 0 \forall t = 1, \dots, T$ . Zudem ist die Summe  $\sum_{t=1}^T \lambda_t^\varphi = 1$  wegen Gl. (A.4) – Gl. (A.5).

## A.4 Alternativbeweis: Minimum zweier strikt quasikonkaver Funktionen

Wir stellen zunächst zwei äquivalente Definitionen von strikter Quasikonkavität voran, die wir für den Beweis benötigen werden.

**Definition 14** (in Anlehnung an Avriel u. a. (2010, S. 57)). Sei  $C \subset \mathbb{R}^n$  eine konvexe Menge. Dann ist eine Funktion  $h : C \rightarrow \mathbb{R}$  strikt quasikonkav genau dann, wenn aus  $\mathbf{x}', \mathbf{x}'' \in C$  und  $0 < \theta < 1$  folgt, dass

$$\min \{h(\mathbf{x}'), h(\mathbf{x}'')\} < h((1 - \theta) \cdot \mathbf{x}' + \theta \cdot \mathbf{x}'').$$

**Definition 15** (in Anlehnung an Mangasarian (1969, S. 132)). Sei  $C \subset \mathbb{R}^n$  eine konvexe Menge. Dann ist eine Funktion  $h : C \rightarrow \mathbb{R}$  strikt quasikonkav genau dann, wenn aus  $\mathbf{x}', \mathbf{x}'' \in C$  mit  $h(\mathbf{x}') < h(\mathbf{x}'')$  und  $0 < \theta < 1$  folgt, dass

$$h(\mathbf{x}') < ((1 - \theta) \cdot \mathbf{x}' + \theta \cdot \mathbf{x}'').$$

Wir wollen nun folgenden Satz beweisen:



**Satz 16.** Sei  $C \subset \mathbb{R}^n$  eine konvexe Menge und seien  $\phi_1(\mathbf{x})$  und  $\phi_2(\mathbf{x})$  zwei strikt quasikonkave Funktionen mit  $\phi_1, \phi_2 : C \rightarrow \mathbb{R}$ . Dann ist die Funktion  $h(\mathbf{x}) := \min \{\phi_1(\mathbf{x}), \phi_2(\mathbf{x})\}$  ebenfalls strikt quasikonkav.

*Beweis.* Es ist nach Definition 15 zu zeigen, dass für zwei beliebige Punkte  $\mathbf{x}'$  und  $\mathbf{x}''$ , die auf einer konvexen Menge definiert sind, aus

$$h(\mathbf{x}') := \min \{\phi_1(\mathbf{x}'), \phi_2(\mathbf{x}')\} < \min \{\phi_1(\mathbf{x}''), \phi_2(\mathbf{x}'')\} =: h(\mathbf{x}'')$$

folgt, dass für  $0 < \theta < 1$

$$\begin{aligned} h(\mathbf{x}') &= \min \{\phi_1(\mathbf{x}'), \phi_2(\mathbf{x}')\} \\ &< \min \{\phi_1((1-\theta) \cdot \mathbf{x}' + \theta \cdot \mathbf{x}''), \phi_2((1-\theta) \cdot \mathbf{x}' + \theta \cdot \mathbf{x}'')\} \\ &= h((1-\theta) \cdot \mathbf{x}' + \theta \cdot \mathbf{x}''). \end{aligned} \tag{A.7}$$

Sei also  $\min \{\phi_1(\mathbf{x}'), \phi_2(\mathbf{x}')\} < \min \{\phi_1(\mathbf{x}''), \phi_2(\mathbf{x}'')\}$ . Da nach Voraussetzung die Funktionen  $\phi_t$  ( $t = 1, 2$ ) strikt quasikonkav sind, gilt nach Definition 15 für  $0 < \theta < 1$

$$\phi_t(\mathbf{x}') < \phi_t(\mathbf{x}'') \Rightarrow \phi_t(\mathbf{x}') < \phi_t((1-\theta) \cdot \mathbf{x}' + \theta \cdot \mathbf{x}''). \tag{A.8}$$

Man kann aus  $\min \{\phi_1(\mathbf{x}'), \phi_2(\mathbf{x}')\} < \min \{\phi_1(\mathbf{x}''), \phi_2(\mathbf{x}'')\}$  schließen, dass für mindestens ein  $t \in \{1, 2\}$   $\phi_t(\mathbf{x}') < \phi_t(\mathbf{x}'')$  gilt. Wir nehmen o. B. d. A. an, dass  $\phi_1(\mathbf{x}') < \phi_1(\mathbf{x}'')$  und betrachten eine Fallunterscheidung für  $\phi_2$ , also 1)  $\phi_2(\mathbf{x}') < \phi_2(\mathbf{x}'')$  und 2)  $\phi_2(\mathbf{x}') \geq \phi_2(\mathbf{x}'')$ .

1) Sei  $\phi_2(\mathbf{x}') < \phi_2(\mathbf{x}'')$ . Für diesen Fall ist Gl. (A.8) für beide  $t$  erfüllt und somit für  $0 < \theta < 1$

$$\min \{\phi_1(\mathbf{x}'), \phi_2(\mathbf{x}')\} < \min \{\phi_1((1-\theta) \cdot \mathbf{x}' + \theta \cdot \mathbf{x}''), \phi_2((1-\theta) \cdot \mathbf{x}' + \theta \cdot \mathbf{x}'')\}.$$

Für diesen Fall ist also auch Gl. (A.7) erfüllt.

2) Sei nun  $\phi_2(\mathbf{x}') \geq \phi_2(\mathbf{x}'')$ , und es gelte weiterhin  $\phi_1(\mathbf{x}') < \phi_1(\mathbf{x}'')$ .

Da  $\min \{\phi_1(\mathbf{x}'), \phi_2(\mathbf{x}')\} < \min \{\phi_1(\mathbf{x}''), \phi_2(\mathbf{x}'')\}$  gilt, ist

$$\phi_1(\mathbf{x}') = \min \{\phi_1(\mathbf{x}'), \phi_2(\mathbf{x}')\} < \phi_2(\mathbf{x}'') \leq \phi_2(\mathbf{x}').$$

Zusammen mit Definition 14 bedeutet das für die strikt quasikonkave Funktion  $\phi_2(\mathbf{x})$ , dass

$$\phi_2(\mathbf{x}'') = \min \{ \phi_2(\mathbf{x}'), \phi_2(\mathbf{x}'') \} < \phi_2((1 - \theta) \cdot \mathbf{x}' + \theta \cdot \mathbf{x}'')$$

für  $0 < \theta < 1$  und somit auch

$$\phi_1(\mathbf{x}') < \phi_2((1 - \theta) \cdot \mathbf{x}' + \theta \cdot \mathbf{x}'') .$$

Selbstverständlich gilt ebenso wie schon in Gl. (A.8) festgestellt

$$\phi_1(\mathbf{x}') < \phi_1((1 - \theta) \cdot \mathbf{x}' + \theta \cdot \mathbf{x}'') .$$

Damit gilt für Fall 2) insgesamt

$$\phi_1(\mathbf{x}') = \min \{ \phi_1(\mathbf{x}'), \phi_2(\mathbf{x}') \} < \min \{ \phi_1((1 - \theta) \cdot \mathbf{x}' + \theta \cdot \mathbf{x}''), \phi_2((1 - \theta) \cdot \mathbf{x}' + \theta \cdot \mathbf{x}'') \}$$

für  $0 < \theta < 1$ . □

## A.5 Berechnung von Importanzen für die Brückenschaltung

Die einzelnen Importanzen  $I_i(\mathbf{p}) = \frac{\partial}{\partial p_i} \varphi(\mathbf{p})$  für eine fünfkomponentige Brückenschaltung berechnen sich wie folgt:

$$\begin{aligned} I_1(\mathbf{p}) &= p_3 + p_4 p_5 - p_2 p_3 p_4 - p_2 p_3 p_5 - p_2 p_4 p_5 - p_3 p_4 p_5 + 2 \cdot \frac{\prod_{i=1}^5 p_i}{p_1} \\ I_2(\mathbf{p}) &= p_4 + p_3 p_5 - p_1 p_3 p_4 - p_1 p_3 p_5 - p_1 p_4 p_5 - p_3 p_4 p_5 + 2 \cdot \frac{\prod_{i=1}^5 p_i}{p_2} \\ I_3(\mathbf{p}) &= p_1 + p_2 p_5 - p_1 p_2 p_4 - p_1 p_2 p_5 - p_1 p_4 p_5 - p_2 p_4 p_5 + 2 \cdot \frac{\prod_{i=1}^5 p_i}{p_3} \\ I_4(\mathbf{p}) &= p_2 + p_1 p_5 - p_1 p_2 p_3 - p_1 p_2 p_5 - p_1 p_3 p_5 - p_2 p_3 p_5 + 2 \cdot \frac{\prod_{i=1}^5 p_i}{p_4} \\ I_5(\mathbf{p}) &= p_2 p_3 + p_1 p_4 - p_1 p_2 p_3 - p_1 p_2 p_4 - p_1 p_3 p_4 - p_2 p_3 p_4 + 2 \cdot \frac{\prod_{i=1}^5 p_i}{p_5} \end{aligned}$$

# B Experimentelle Performance-Analysen

## B.1 Übersichten der Instanzparameter

System	Seriell					System	Parallel				
Instanznr.	1	2	3	4	5	Instanznr.	1	2	3	4	5
$\hat{a}_1$	2,8	4,9	1,9	0,9	1,7	$\hat{a}_1$	3,6	4,0	2,2	0,6	2,1
$\hat{a}_2$	2,2	0,7	3,4	1,1	2,9	$\hat{a}_2$	3,3	3,2	3,4	1,9	1,4
$\hat{a}_3$	4,8	2,7	1,6	4,2	2,9	$\hat{a}_3$	2,0	2,1	1,9	0,6	4,7
$\hat{a}_4$	2,9	2,3	4,9	4,9	0,7	$\hat{a}_4$	4,6	2,9	0,6	3,1	3,6
$\hat{a}_5$	0,2	3,1	3,3	1,5	3,0	$\hat{a}_5$	4,2	3,1	1,3	0,7	0,2

(a) Seriensystem

(b) Parallelsystem

System	Serien-Parallel					System	Brücke, 5 Komp.				
Instanznr.	1	2	3	4	5	Instanznr.	1	2	3	4	5
$\hat{a}_1$	3,1	2,4	3,5	3,3	0,4	$\hat{a}_1$	2,8	3,9	0,4	0,8	3,5
$\hat{a}_2$	2,2	0,8	1,1	4,1	2,7	$\hat{a}_2$	4,5	4,7	0,7	0,6	0,6
$\hat{a}_3$	4,0	1,4	2,9	4,9	1,7	$\hat{a}_3$	2,7	3,9	2,2	2,4	3,4
$\hat{a}_4$	0,4	0,2	1,1	3,7	1,5	$\hat{a}_4$	0,5	3,1	0,9	0,3	1,0
$\hat{a}_5$	3,9	4,7	2,4	1,6	4,2	$\hat{a}_5$	4,4	3,4	1,3	0,7	0,1

(c) Serien-Parallel-System

(d) Brückenschaltung (5 Komp.)

System	Brücke, 8 Komp.				
Instanznr.	1	2	3	4	5
$\hat{a}_1$	3,4	1,2	4,4	2,2	0,8
$\hat{a}_2$	1,4	4,8	2,9	3,1	1,6
$\hat{a}_3$	3,7	2,3	4,4	0,2	2,9
$\hat{a}_4$	4,8	4,2	3,5	1,6	3,0
$\hat{a}_5$	1,3	3,5	3,7	3,3	2,9
$\hat{a}_6$	2,9	1,6	2,6	1,5	3,3
$\hat{a}_7$	3,0	4,1	4,8	3,1	3,3
$\hat{a}_8$	2,9	2,0	3,3	2,2	2,2

(e) Brückenschaltung (8 Komp.)

Tab. B.1: Parameterkonfiguration der einzelnen Komponenten – Skalierungsparameter für die Investitionsfunktion, gerundet auf eine Nachkommastelle  
Quelle: Eigene Darstellung

System Instanznr.	Seriiell					Parallel					Serien-Parallel					Brücke, 5 Komp.				
	1	2	3	4	5	1	2	3	4	5	1	2	3	4	5	1	2	3	4	5
$\hat{u}_1$	16,2	17,3	15,4	17,3	19,6	67,0	65,9	64,7	65,4	63,4	15,3	18,5	16,5	17,0	15,8	18,6	16,4	17,9	19,5	18,1
$\hat{u}_2$	15,1	16,9	19,8	18,5	17,0	76,3	79,8	73,8	67,6	69,0	15,7	18,9	15,2	16,6	17,1	15,4	15,3	17,2	18,8	16,3
$\hat{u}_3$	15,4	18,5	17,6	18,2	19,2	79,7	65,0	61,2	67,5	66,1	16,2	18,9	18,4	15,8	19,8	17,1	16,3	16,4	19,0	16,8
$\hat{u}_4$	17,8	16,9	15,6	19,8	16,8	79,4	62,1	74,6	75,0	76,8	18,6	16,4	19,8	18,0	15,9	16,4	16,9	15,9	19,3	17,0
$\hat{u}_5$	15,7	16,0	19,5	17,3	19,6	78,1	79,0	77,6	64,8	64,8	16,5	15,2	17,3	17,9	19,7	16,3	19,0	19,4	19,1	19,3
$\hat{c}_1$	0,9	0,6	0,9	0,8	1,1	0,8	0,7	0,9	0,9	0,6	1,2	0,9	0,7	1,0	0,9	0,7	0,9	0,7	0,6	0,8
$\hat{c}_2$	0,7	1,1	1,1	1,1	0,9	0,8	1,2	0,6	1,0	0,8	1,0	0,7	0,9	1,0	0,7	1,0	0,7	0,7	0,9	0,6
$\hat{c}_3$	0,8	1,0	0,9	0,7	0,9	1,2	1,0	1,2	0,8	1,1	0,8	1,1	0,9	0,8	0,8	1,0	1,2	1,1	0,8	0,7
$\hat{c}_4$	1,0	1,2	0,7	0,6	0,6	1,1	0,8	0,8	1,0	0,7	1,2	0,8	0,8	0,7	0,7	1,1	0,7	0,8	1,0	0,7
$\hat{c}_5$	1,1	1,0	1,0	1,0	0,8	1,0	1,0	0,8	1,0	0,6	0,8	1,1	1,0	1,2	0,6	0,7	0,7	1,0	0,7	1,1
$\hat{b}_1$	1,3	1,1	1,1	1,0	1,4	1,5	1,3	1,2	1,2	1,0	1,3	1,4	1,4	1,3	1,2	1,4	1,2	1,3	1,1	1,3
$\hat{b}_2$	1,5	1,3	1,3	1,1	1,3	1,4	1,1	1,4	1,1	1,0	1,3	1,2	1,1	1,3	1,1	1,5	1,2	1,2	1,3	1,2
$\hat{b}_3$	1,5	1,1	1,3	1,3	1,3	1,2	1,3	1,4	1,5	1,4	1,0	1,2	1,3	1,0	1,3	1,1	1,2	1,1	1,3	1,1
$\hat{b}_4$	1,1	1,3	1,0	1,3	1,5	1,2	1,4	1,4	1,3	1,4	1,1	1,2	1,0	1,5	1,3	1,3	1,5	1,2	1,0	1,5
$\hat{b}_5$	1,2	1,4	1,3	1,3	1,3	1,4	1,0	1,5	1,4	1,5	1,5	1,1	1,0	1,3	1,4	1,3	1,2	1,2	1,2	1,5

Tab. B.2: Parameterkonfiguration der einzelnen Komponenten für die fünfkompontigen Systeme – Form- und Skalierungsparameter für die Verschleißfunktion, gerundet auf eine Nachkommastelle

Quelle: Eigene Darstellung

System	Brücke, 8 Komp.				
Instanznr.	1	2	3	4	5
$\hat{u}_1$	17,3	16,3	20,0	16,4	16,4
$\hat{u}_2$	16,6	16,1	17,0	16,2	18,5
$\hat{u}_3$	16,2	16,3	15,8	18,3	17,1
$\hat{u}_4$	18,1	15,7	18,2	16,2	16,8
$\hat{u}_5$	15,2	15,1	17,5	18,3	19,1
$\hat{u}_6$	15,1	15,6	19,9	17,6	19,6
$\hat{u}_7$	17,1	18,1	15,3	17,1	15,2
$\hat{u}_8$	15,3	19,9	18,9	17,8	16,2
$\hat{c}_1$	0,7	0,8	1,0	0,9	0,6
$\hat{c}_2$	0,9	1,1	1,2	1,0	0,7
$\hat{c}_3$	1,1	1,1	1,1	0,8	1,0
$\hat{c}_4$	0,9	1,2	0,8	0,9	0,6
$\hat{c}_5$	0,7	0,6	0,9	0,7	1,1
$\hat{c}_6$	0,7	1,1	0,8	0,7	1,0
$\hat{c}_7$	0,9	0,8	0,8	0,6	1,2
$\hat{c}_8$	0,9	1,1	0,8	1,2	0,9
$\hat{b}_1$	1,5	1,2	1,0	1,2	1,3
$\hat{b}_2$	1,0	1,5	1,0	1,5	1,0
$\hat{b}_3$	1,1	1,1	1,0	1,2	1,3
$\hat{b}_4$	1,2	1,3	1,5	1,2	1,2
$\hat{b}_5$	1,5	1,2	1,4	1,2	1,1
$\hat{b}_6$	1,2	1,0	1,4	1,5	1,1
$\hat{b}_7$	1,1	1,3	1,3	1,1	1,3
$\hat{b}_8$	1,4	1,1	1,3	1,2	1,2

Tab. B.3: Parameterkonfiguration der einzelnen Komponenten für das achtkomponentige System – Form- und Skalierungsparameter für die Verschleißfunktion, gerundet auf eine Nachkommastelle

Quelle: Eigene Darstellung

System	Seriell				
Instanznr.	1	2	3	4	5
$\widehat{p}_1$	0,88	0,84	0,73	0,76	0,83
$\widehat{p}_2$	0,86	0,90	0,89	0,70	0,73
$\widehat{p}_3$	0,84	0,85	0,85	0,81	0,89
$\widehat{p}_4$	0,73	0,74	0,74	0,83	0,80
$\widehat{p}_5$	0,86	0,77	0,75	0,76	0,77

(a) Seriensystem

System	Parallel				
Instanznr.	1	2	3	4	5
$\widehat{p}_1$	0,13	0,11	0,14	0,12	0,19
$\widehat{p}_2$	0,18	0,20	0,11	0,11	0,18
$\widehat{p}_3$	0,16	0,15	0,18	0,19	0,13
$\widehat{p}_4$	0,10	0,15	0,12	0,20	0,11
$\widehat{p}_5$	0,13	0,16	0,15	0,20	0,14

(b) Parallelsystem

System	Serien-Parallel				
Instanznr.	1	2	3	4	5
$\widehat{p}_1$	0,79	0,84	0,82	0,76	0,75
$\widehat{p}_2$	0,72	0,70	0,81	0,84	0,84
$\widehat{p}_3$	0,73	0,80	0,78	0,90	0,75
$\widehat{p}_4$	0,72	0,86	0,85	0,70	0,74
$\widehat{p}_5$	0,78	0,84	0,71	0,82	0,76

(c) Serien-Parallel-System

System	Brücke, 5 Komp.				
Instanznr.	1	2	3	4	5
$\widehat{p}_1$	0,88	0,87	0,75	0,74	0,86
$\widehat{p}_2$	0,88	0,77	0,82	0,79	0,74
$\widehat{p}_3$	0,87	0,71	0,74	0,74	0,74
$\widehat{p}_4$	0,87	0,74	0,80	0,85	0,78
$\widehat{p}_5$	0,71	0,71	0,79	0,87	0,82

(d) Brückenschaltung (5 Komp.)

System	Brücke, 8 Komp.				
Instanznr.	1	2	3	4	5
$\widehat{p}_1$	0,87	0,73	0,81	0,83	0,77
$\widehat{p}_2$	0,72	0,80	0,82	0,86	0,89
$\widehat{p}_3$	0,80	0,77	0,89	0,76	0,75
$\widehat{p}_4$	0,73	0,89	0,76	0,82	0,89
$\widehat{p}_5$	0,84	0,85	0,75	0,71	0,88
$\widehat{p}_6$	0,78	0,85	0,72	0,80	0,86
$\widehat{p}_7$	0,81	0,88	0,70	0,90	0,83
$\widehat{p}_8$	0,74	0,72	0,89	0,88	0,87

(e) Brückenschaltung (8 Komp.)

Tab. B.4: Parameterkonfiguration der einzelnen Komponenten – Startwerte für den prozentualen Abnutzungsvorrat, gerundet auf zwei Nachkommastellen  
Quelle: Eigene Darstellung

System	Seriell				
Instanznr.	1	2	3	4	5
$\widehat{p}_1$	0,88	0,84	0,73	0,76	0,83
$\widehat{p}_2$	0,86	0,90	0,89	0,70	0,73
$\widehat{p}_3$	0,84	0,85	0,85	0,81	0,89
$\widehat{p}_4$	0,73	0,74	0,74	0,83	0,80
$\widehat{p}_5$	0,86	0,77	0,75	0,76	0,77

(a) Seriensystem

System	Parallel				
Instanznr.	1	2	3	4	5
$\widehat{p}_1$	0,19	0,14	0,21	0,17	0,37
$\widehat{p}_2$	0,35	0,39	0,14	0,13	0,33
$\widehat{p}_3$	0,29	0,24	0,35	0,36	0,20
$\widehat{p}_4$	0,10	0,25	0,16	0,39	0,12
$\widehat{p}_5$	0,20	0,29	0,25	0,39	0,22

(b) Parallelsystem

System	Serien-Parallel				
Instanznr.	1	2	3	4	5
$\widehat{p}_1$	0,79	0,84	0,82	0,76	0,75
$\widehat{p}_2$	0,72	0,70	0,81	0,84	0,84
$\widehat{p}_3$	0,73	0,80	0,78	0,90	0,75
$\widehat{p}_4$	0,72	0,86	0,85	0,70	0,74
$\widehat{p}_5$	0,78	0,84	0,71	0,82	0,76

(c) Serien-Parallel-System

System	Brücke, 5 Komp.				
Instanznr.	1	2	3	4	5
$\widehat{p}_1$	0,88	0,87	0,75	0,74	0,86
$\widehat{p}_2$	0,88	0,77	0,82	0,79	0,74
$\widehat{p}_3$	0,87	0,71	0,74	0,74	0,74
$\widehat{p}_4$	0,87	0,74	0,80	0,85	0,78
$\widehat{p}_5$	0,71	0,71	0,79	0,87	0,82

(d) Brückenschaltung (5 Komp.)

System	Brücke, 8 Komp.				
Instanznr.	1	2	3	4	5
$\widehat{p}_1$	0,87	0,73	0,81	0,83	0,77
$\widehat{p}_2$	0,72	0,80	0,82	0,86	0,89
$\widehat{p}_3$	0,80	0,77	0,89	0,76	0,75
$\widehat{p}_4$	0,73	0,89	0,76	0,82	0,89
$\widehat{p}_5$	0,84	0,85	0,75	0,71	0,88
$\widehat{p}_6$	0,78	0,85	0,72	0,80	0,86
$\widehat{p}_7$	0,81	0,88	0,70	0,90	0,83
$\widehat{p}_8$	0,74	0,72	0,89	0,88	0,87

(e) Brückenschaltung (8 Komp.)

Tab. B.5: Parameterkonfiguration der einzelnen Komponenten – Startwerte für die Komponentenleistung, gerundet auf zwei Nachkommastellen  
Quelle: Eigene Darstellung



## **B.2 Ausgewählter Quellcode**

In diesem Anhangsteil findet sich Programmcode für GAMS (Abschnitt B.2.1) und für Python (Abschnitte B.2.2 bis B.2.3). Für den Python-Code gilt, dass die Nummerierung der Instanzen bei 0 statt bei 1 beginnt und entsprechend bis 4 statt 5 (bzw. 7 statt 8) läuft.

### **B.2.1 Modellierung in GAMS**

Der Optimierungssolver CONOPT wurde über die Modellierungssprache GAMS (s. GAMS Development Corporation (2020)) angesteuert. Für eine Beispiel-Instanz ist der Code im folgenden dargestellt.

```

1 $title Periodenübergreifende Budgetallokation in der Instandhaltung modularer»
  Systeme
2
3 Sets
4     i      Komponenten / i1*i5 /
5     t      Perioden   /t0*t5/
6
7 Scalar
8     B      Budget
9     PH     Länge des Planungshorizonts ;
10
11 PH=card(t)-1;
12
13 * Pro Periode 1 Geldeinheit
14 B=PH;
15
16 Parameters
17     p0(i)  Anfangsleistung
18           /
19 i1        0.5
20 i2        0.6
21 i3        0.7
22 i4        0.8
23 i5        0.6
24 /
25
26     a(i)  Skalierungsparameter für Investitionsfunktion
27           /
28 i1        2
29 i2        4
30 i3        3
31 i4        2
32 i5        1
33 /
34
35 deltaW(i)  Verschleiß von i in Periode t
36 /
37 i1        0.15
38 i2        0.1
39 i3        0.2
40 i4        0.3
41 i5        0.2
42 /
43
44
45 Variables
46     Bt(t)  Rest-Budget am Anfang von Periode t
47     phi(t) Wert der Strukturfunktion zu Beginn von Periode t
48     x(i,t) Investition in Komponente i zu Beginn von Periode t
49     p(i,t) Zustand von Komponente i zu Beginn von Periode t
50     gS     Minimum aller periodenbezogenen Strukturfunktionswerte
51     z      Zielfunktionswert
52
53 Positive Variable x, Bt, phi ;
54
55
56 *Hilfe für den Solver
57 phi.up(t)=1.0;
58
59 *Initialisierung
60 Bt.fx("t0")=B;
61 p.fx(i,"t0")= p0(i);
62
63

```

```

64 Equations
65     ZFW                                     Berechnung des Zielfunktionswertes f»
    ür die Minimumfunktion
66     Strukturfunktion(t)                   Berechnung der Strukturfunktion (Brü»
    ckenschaltung) zu Beginn der Periode t
67     Budgetrestriktion
68     Budgetentwicklung(t)
69     Komponentenentwicklung(i,t)
70     Hilfsnebenbedingung(t)               Berechnung des Minimums der perioden»
    bezogenen Strukturfunktionswerte
71 ;
72
73 *Strukturfunktion(t)..                    phi(t) =e= p("i1",t)*p("i2",t) *»
    p("i3",t) * p("i4",t) * p("i5",t) ;
74 *Strukturfunktion(t)..                    phi(t) =e= p("i5",t)*(1-(1-p("i1",»
    t))*(1-p("i2",t)))*(1-(1-p("i3",t)))*(1-p("i4",t)))+(1-p("i5",t))*(1-(1-p("i1"»
    ,t)*p("i3",t))*(1-p("i2",t)*p("i4",t)));
75 *Strukturfunktion(t)..                    phi(t) =e= 1 - (1- p("i1",t))*(1-»
    p("i2",t)) * (1- p("i3",t)) * (1-p("i4",t)) * (1- p("i5",t)) ;
76 Strukturfunktion(t)..                    phi(t) =e= 1 - (1- p("i1",t)*p("i2»
    ",t)* p("i3",t)) * (1-p("i4",t) * p("i5",t)) ;
77 ZFW..                                     z =e= gS ;
78 Hilfsnebenbedingung(t)$(ord(t)>1)..      gS =1= phi(t);
79 Budgetrestriktion..                      sum((i,t),x(i,t)) =e= B;
80 Budgetentwicklung(t)$(ord(t)<=PH)..      Bt(t+1)=e=Bt(t)-sum(i,x(i,t));
81 Komponentenentwicklung(i,t)$(ord(t)<=PH).. p(i,t+1) =e= 1-(1-p(i,t))*exp(-a»
    (i)*x(i,t))-deltaW(i) ;
82
83 Model maintenance /all/ ;
84
85 option optca=0, optcr=0, reslim=100, nlp=lindoglobal;
86
87 Solve maintenance using nlp maximizing z ;
88
89 Display x.l, Bt.l, z.l, phi.l ;
90

```

## **B.2.2 Generierung der gerechneten Instanzen**

---

```
createInstances.py
```

---

```

1  import readwrite as rw
2  import numpy as np
3  import os
4  import os.path
5
6
7  if __name__ == "__main__":
8      np.random.seed(0) # Zahl in der Klammer
                           definiert Instanzen
9
10     current_directory = os.getcwd()
11     final_directory = os.path.join(current_directory,
                                     r'Instances')
12
13     if not os.path.exists(final_directory):
14         os.makedirs(final_directory)
15
16     # Grenzwerte: [0]:unten, [1]:oben
17     a_lim, p_lim, u_lim, b_lim, c_lim, a, p, u, b, c =
18     [
19     ], [], [], [], [], [], [], [], [], []
20     a_lim = [0.1, 5]
21     p_lim = [0.7, 0.9]
22     p_par_lim = [0.1, 0.2]
23     u_lim = [15, 20]
24     u_par_lim = [60, 80]
25     b_lim = [1, 1.5]
26     c_lim = [0.6, 1.2]
27     # c_ser_lim = [0.6, 1.2]
28
29     nb_config_5 = 4 # Seriell, Parallel,
                       Serien-Parallel, Brückenschaltung mit je 5
                       Komponenten
30     nb_config_8 = 1 # Brückenschaltung mit 8
                       Komponenten
31     nb_inst_per_config = 5
32
33     size = nb_inst_per_config*(nb_config_5*5 +
34                                nb_config_8*8)
35
36     a = np.random.uniform(low=a_lim[0], high=a_lim[1],
37                            size=size)
38     p = np.random.uniform(low=p_lim[0], high=p_lim[1],
39                            size=size)
40     p_par = np.random.uniform(
41         low=p_par_lim[0], high=p_par_lim[1], size=
42         nb_inst_per_config*5)
43     # c_ser = np.random.uniform(
44     #     low=c_ser_lim[0], high=c_ser_lim[1],

```

---

*createInstances.py*

---

```

size=nb_inst_per_config*5)
40 u = np.random.uniform(low=u_lim[0], high=u_lim[1],
    size=size)
41 u_par = np.random.uniform(low=u_par_lim[0], high=
u_par_lim[1], size=nb_inst_per_config*5)
42
43 b = np.random.uniform(low=b_lim[0], high=b_lim[1],
    size=size)
44 c = np.random.uniform(low=c_lim[0], high=c_lim[1],
    size=size)
45
46 a5 = np.array(a[:nb_inst_per_config*nb_config_5*5]
47               ).reshape(nb_inst_per_config, 5,
    nb_config_5)
48 p5 = np.array(p[:nb_inst_per_config*nb_config_5*5]
49               ).reshape(nb_inst_per_config, 5,
    nb_config_5)
50 p5_par = np.array(p_par).reshape(
nb_inst_per_config, 5)
51 # c5_ser =
np.array(c_ser).reshape(nb_inst_per_config, 5)
52 u5 = np.array(u[:nb_inst_per_config*nb_config_5*5]
53               ).reshape(nb_inst_per_config, 5,
    nb_config_5)
54 u5_par = np.array(u_par).reshape(
nb_inst_per_config, 5)
55 b5 = np.array(b[:nb_inst_per_config*nb_config_5*5]
56               ).reshape(nb_inst_per_config, 5,
    nb_config_5)
57 c5 = np.array(c[:nb_inst_per_config*nb_config_5*5]
58               ).reshape(nb_inst_per_config, 5,
    nb_config_5)
59
60 a8 = np.array(
61     a[nb_inst_per_config*nb_config_5*5:]).reshape(
    nb_inst_per_config, 8)
62 p8 = np.array(
63     p[nb_inst_per_config*nb_config_5*5:]).reshape(
    nb_inst_per_config, 8)
64 u8 = np.array(
65     u[nb_inst_per_config*nb_config_5*5:]).reshape(
    nb_inst_per_config, 8)
66 b8 = np.array(
67     b[nb_inst_per_config*nb_config_5*5:]).reshape(
    nb_inst_per_config, 8)
68 c8 = np.array(
69     c[nb_inst_per_config*nb_config_5*5:]).reshape(
    nb_inst_per_config, 8)
70

```

createInstances.py

```

71     structures = ["seriell", "parallel", "ser_par",
72                  "bruecke"]
73     final_directory = final_directory+"/"
74     i = 0
75     for s in structures:
76         for j in range(nb_inst_per_config):
77             path = final_directory+s+"_"+str(j)+"_a5"
78             rw.write_instance(path, a5[j, :, i])
79             path = final_directory+s+"_"+str(j)+"_p5"
80             rw.write_instance(path, p5[j, :, i])
81             path = final_directory+s+"_"+str(j)+"_u5"
82             rw.write_instance(path, u5[j, :, i])
83             path = final_directory+s+"_"+str(j)+"_b5"
84             rw.write_instance(path, b5[j, :, i])
85             path = final_directory+s+"_"+str(j)+"_c5"
86             rw.write_instance(path, c5[j, :, i])
87             i += 1
88
89     # Überschreibe Startleistungen für
90     # Parallelschaltung, um sinnvollere
91     # Beispiel-Instanzen zu erhalten
92     s = "parallel"
93
94     for j in range(nb_inst_per_config):
95         path = final_directory+s+"_"+str(j)+"_p5"
96         rw.write_instance(path, p5_par[j, :])
97         path = final_directory+s+"_"+str(j)+"_u5"
98         rw.write_instance(path, u5_par[j, :])
99
100     # Überschreibe Verschleißparameter für
101     # Serienschaltung, um sinnvollere
102     # Beispiel-Instanzen zu erhalten
103     # s = "seriell"
104
105     # for j in range(nb_inst_per_config):
106     #     path = final_directory+s+"_"+str(j)+"_c5"
107     #     rw.write_instance(path, c5_ser[j, :])
108
109     # Sonderfall 8-komponentige Brückenschaltung
110     s = "bruecke"
111
112     for j in range(nb_inst_per_config):
113         path = final_directory+s+"_"+str(j)+"_a8"
114         rw.write_instance(path, a8[j, :])
115         path = final_directory+s+"_"+str(j)+"_p8"
116         rw.write_instance(path, p8[j, :])
117         path = final_directory+s+"_"+str(j)+"_u8"
118         rw.write_instance(path, u8[j, :])

```

*createInstances.py*

---

```
115         path = final_directory+s+"_"+str(j)+"_b8"
116         rw.write_instance(path, b8[j, :])
117         path = final_directory+s+"_"+str(j)+"_c8"
118         rw.write_instance(path, c8[j, :])
```



### **B.2.3 Einflussanalyse mit Lasso-Regression**

Einlesen der Instanzparameter

---

```
read_inst_from_csv.py
```

```

1  import pandas as pd
2  import readwrite as rw
3
4  import os
5
6
7  if __name__ == "__main__":
8      current_directory = os.getcwd()
9      pre_path = os.path.join(current_directory,
10                               r'Instances')
11     structures = ["bruecke", "parallel", "seriell",
12                  "ser_par"]
13     var_names = ["a", "b", "c", "p", "u"]
14     nb_of_instances = 5
15     nb_of_components = 5
16     instances = []
17     inst_comb = []
18     for cur_struc in structures:
19         for instance_counter in range(nb_of_instances):
20             for var_name in var_names:
21                 instance_path = pre_path+"\\\" + \
22                               cur_struc + "_" + str(
23                                   instance_counter)
24                 instances.append(rw.read_instance(
25                     instance_path, var_name,
26                     nb_of_components))
27                 inst_comb.append([cur_struc, str(
28                     instance_counter), var_name])
29     for instance_counter in range(nb_of_instances):
30         for var_name in var_names:
31             instance_path = pre_path+"\\\" + \
32                             "bruecke" + "_" + str(instance_counter)
33             instances.append(rw.read_instance(
34                 instance_path, var_name, 8))
35             inst_comb.append(["bruecke", str(
36                 instance_counter), var_name])
37     df = pd.DataFrame.from_records(instances)
38     df2 = pd.DataFrame.from_records(inst_comb)
39     df3 = df.merge(df2, left_index=True, right_index=
40                   True)
41
42     with pd.ExcelWriter("instance_params.xls") as
43         writer:
44             df3.to_excel(writer)

```

## **Bestimmung der Koeffizienten der Regression**

*lassoLARS.py*

---

```
1  import pandas as pd
2
3  from sklearn.linear_model import LassoLarsCV
4
5
6  if __name__ == "__main__":
7
8      df = pd.read_excel("results.xls")
9
10     X = df[["aMittel", "aStd", "bMittel", "bStd",
11            "cMittel", "cStd", "pMittel", "pStd",
12            "uMittel", "uStd", "bruecke", "seriell",
13            "parallel", "ser_par",
14            "AnzKomp"]]
15     y = df["prozErg"]
16
17     reg = LassoLarsCV().fit(X, y)
18
19     print(reg.score(X, y))
20     print(reg.coef_)
21     print(reg.intercept_)
```

## B.3 Reproduzierbarkeit

Um die Reproduzierbarkeit der Ergebnisse zu gewährleisten, wurden diverse Anstrengungen unternommen.

- Es wurden Seeds gesetzt, sowohl für die Pakete numpy [`numpy.random.seed()`] und TensorFlow [`tensorflow.set_random_seed()`] als auch für das Zufalls-Verhalten von Python [`os.environ['PYTHONHASHSEED']` bzw. `random.seed()`].
- Getestet wurde zudem die Anwendung einer Funktion, die deterministisches Verhalten von TensorFlow herbeiführen soll [`tfdeterminism.patch()`].

- Zuletzt wurde die Anzahl der genutzten CPU-Kerne auf 1 reduziert, um Interaktionen, die zu stochastischem Verhalten führen können, auszuschließen:

```
config = tensorflow.ConfigProto(intra_op_parallelism_threads=1,
inter_op_parallelism_threads=1, device_count={'CPU': 1})
session = tensorflow.Session(config=config)
tensorflow.backend.set_session(session)
```

- Bei den verrauschten Netzen wurde zwischen Lern- und Testphase unterschieden, sodass beim Testen die stochastische Schicht des Netzes nur noch deterministisch arbeitet.

Die Einstellungen führten zwar dazu, dass die Generierung der Instanzen reproduzierbar durchgeführt wurde, das Deep-Reinforcement-Learning-Verfahren allerdings weiterhin nicht-deterministisches Verhalten aufwies.



## B.4 Liste installierter Python-Pakete

absl-py==0.11.0	google-pasta==0.2.0	opt-einsum==3.1.0	scipy==1.5.2
argon2-cffi==20.1.0	grpcio==1.31.0	optcc==4	Send2Trash==1.5.0
astor==0.8.1	h5py==2.10.0	packaging==20.8	setuptools==51.1.2.post20210112
async-generator==1.10	idna==2.10	pandas==0.23.4	six==1.15.0
attrs==20.3.0	idncc==1	pandocfilters==1.4.3	tensorboard==2.3.0
autopep8==1.5.3	importlib-metadata==2.0.0	parso==0.7.0	tensorboard-plugin-wit==1.6.0
backcall==0.2.0	ipykernel==5.3.4	pep8==1.7.1	tensorflow==1.15.0
bleach==3.2.1	ipython==7.19.0	pickleshare==0.7.5	tensorflow-determinism==0.3.0
blinker==1.4	ipython-genutils==0.2.0	pip==20.3.3	tensorflow-estimator==1.15.1
bratlipy==0.7.0	jedi==0.18.0	prometheus-client==0.9.0	termcolor==1.1.0
cachetools==4.2.0	Jinja2==2.11.2	prompt-toolkit==3.0.8	terminado==0.9.2
certifi==2020.12.5	joblib==1.0.0	protobuf==3.13.0	testpath==0.4.4
cffi==1.14.4	jsonschema==3.2.0	pyasn1==0.4.8	threadpoolctl==2.1.0
cfgmcc==4	jupyter-client==6.1.7	pyasn1-modules==0.2.8	toml==0.10.1
chardet==4.0.0	jupyter-core==4.7.0	pycodestyle==2.6.0	tornado==6.1
click==7.1.2	jupyterlab-pygments==0.1.2	pycparser==2.20	traitlets==5.0.5
colorama==0.4.4	Keras==2.1.6	Pygments==2.7.4	urllib3==1.26.2
cryptography==3.3.1	Keras-Applications==1.0.8	PyJWT==1.7.1	wcwidth==0.2.5
cycler==0.10.0	Keras-Preprocessing==1.1.0	pyOpenSSL==20.0.1	webencodings==0.5.1
dctmcc==2	kiwisolver==1.3.0	pyparsing==2.4.7	Werkzeug==0.16.1
decorator==4.4.2	Markdown==3.3.3	pyreadline==2.1	wheel==0.36.2
defusedxml==0.6.0	MarkupSafe==1.1.1	pyrsistent==0.17.3	win-inet-pton==1.1.0
emptyacc==0.0.0	matplotlib==3.2.2	PySocks==1.7.1	wincertstore==0.2
entrypoints==0.3	mistune==0.8.4	python-dateutil==2.8.1	wrapt==1.12.1
GAMS==1.0	mkl-fft==1.2.0	pytz==2020.5	xlrd==2.0.1
gamsemb==0.0.0	mkl-random==1.1.1	pywin32==227	xlwt==1.3.0
gamsxcc==1	mkl-service==2.3.0	pywinpty==0.5.7	zipp==3.4.0
gast==0.2.2	nbclient==0.5.1	PyYAML==5.3.1	
gdxcc==7	nbconvert==6.0.7	pyzmq==20.0.0	
gevmcc==6	nbformat==5.1.2	requests==2.25.1	
gmdec==2	nest-asyncio==1.4.3	requests-oauthlib==1.3.0	
gmomcc==17	notebook==6.1.6	rope==0.17.0	
google-auth==1.17.2	numpy==1.19.2	rsa==4.7	
google-auth-oauthlib==0.4.2	oauthlib==3.1.0	scikit-learn==0.23.2	

## B.5 Weitere Ergebnis-Grafiken

Für die folgenden Grafiken gilt, dass nur jeder zehnte Wert aus Übersichtlichkeitsgründen geplottet wurde.

### B.5.1 Fünfkomponentige Systeme

#### Seriensystem

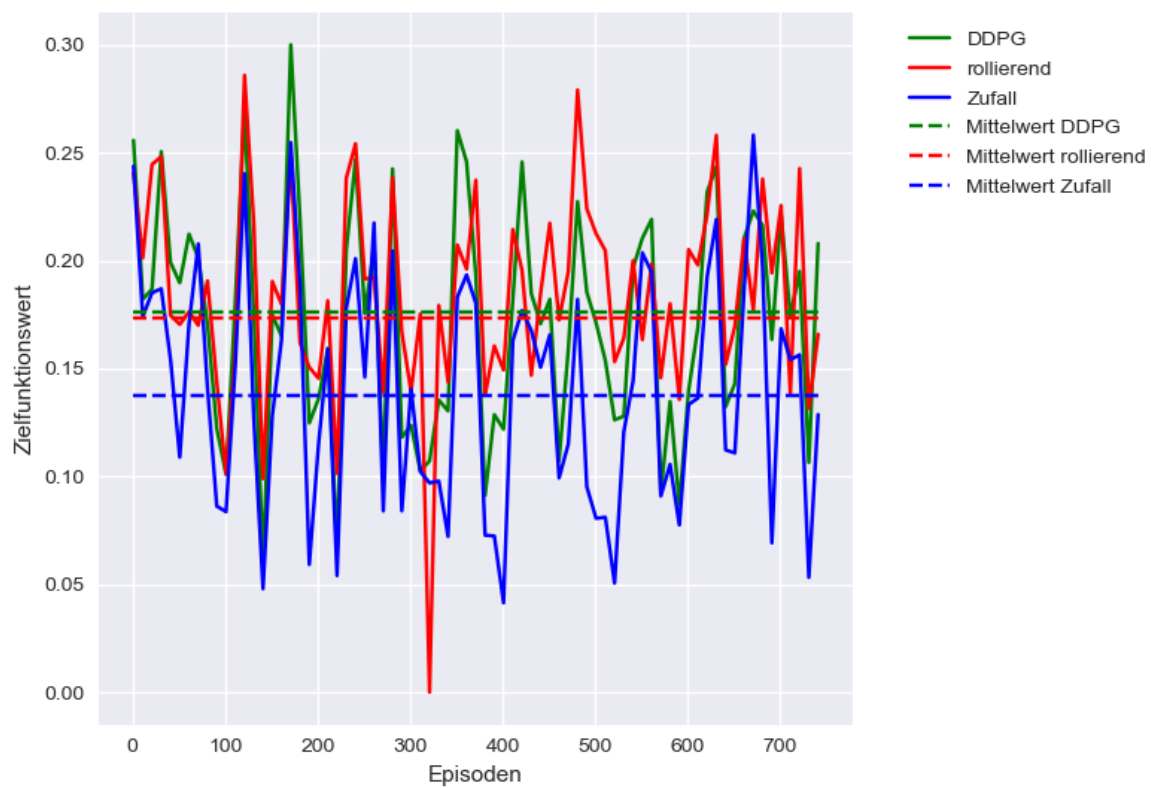


Abb. B.1: Instanz Nr. 2 Seriensystem

Quelle: Eigene Darstellung



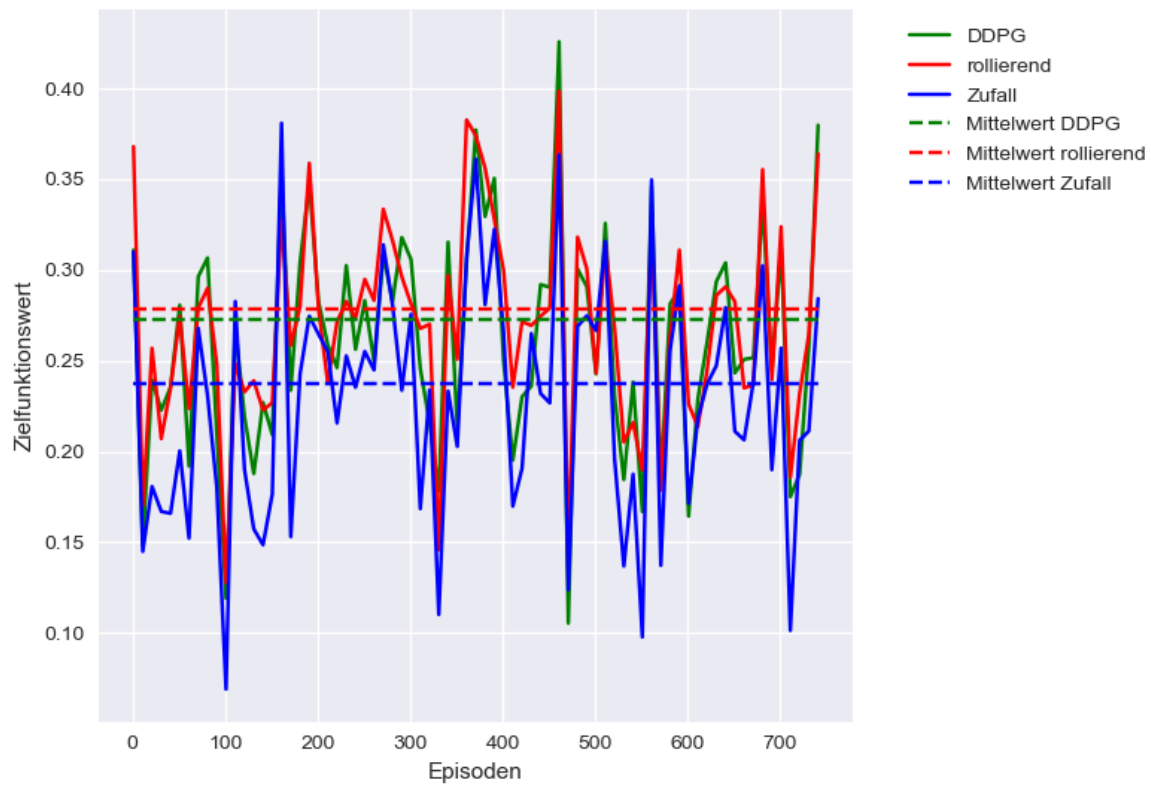


Abb. B.2: Instanz Nr. 3 Seriensystem  
Quelle: Eigene Darstellung

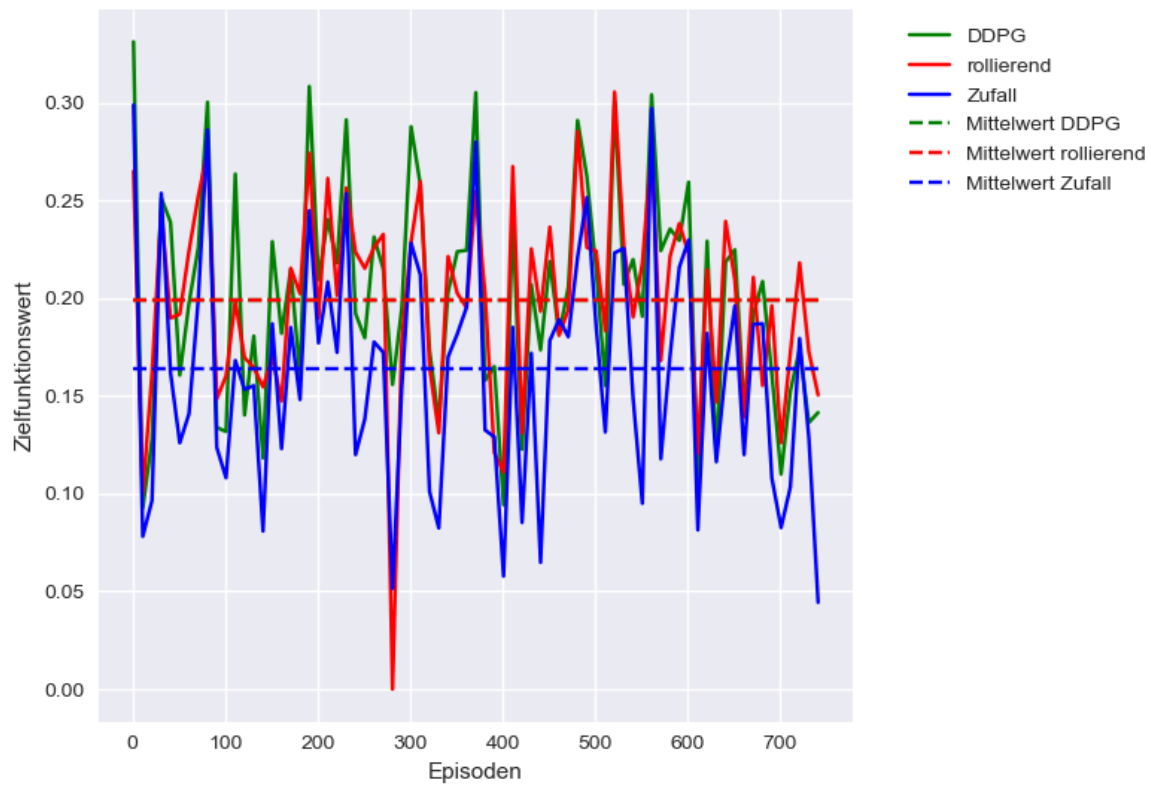


Abb. B.3: Instanz Nr. 4 Seriensystem  
Quelle: Eigene Darstellung

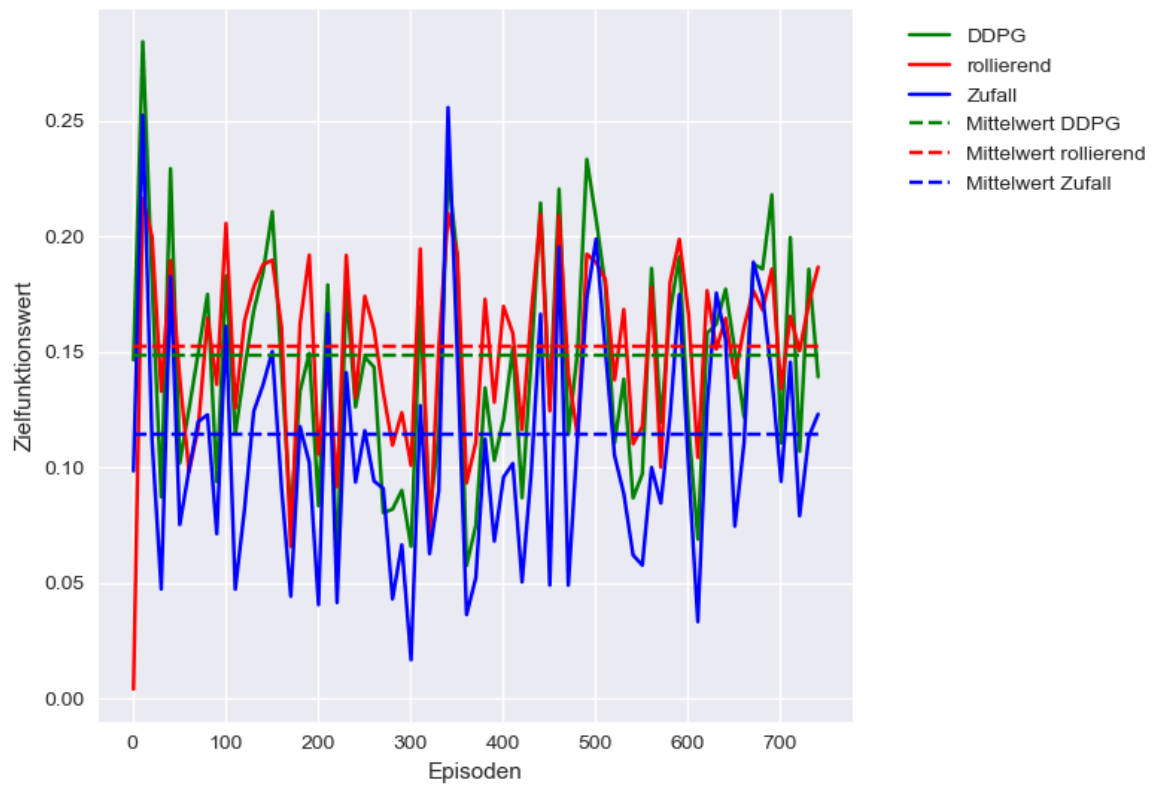


Abb. B.4: Instanz Nr. 5 Seriensystem  
Quelle: Eigene Darstellung

## Parallelsystem

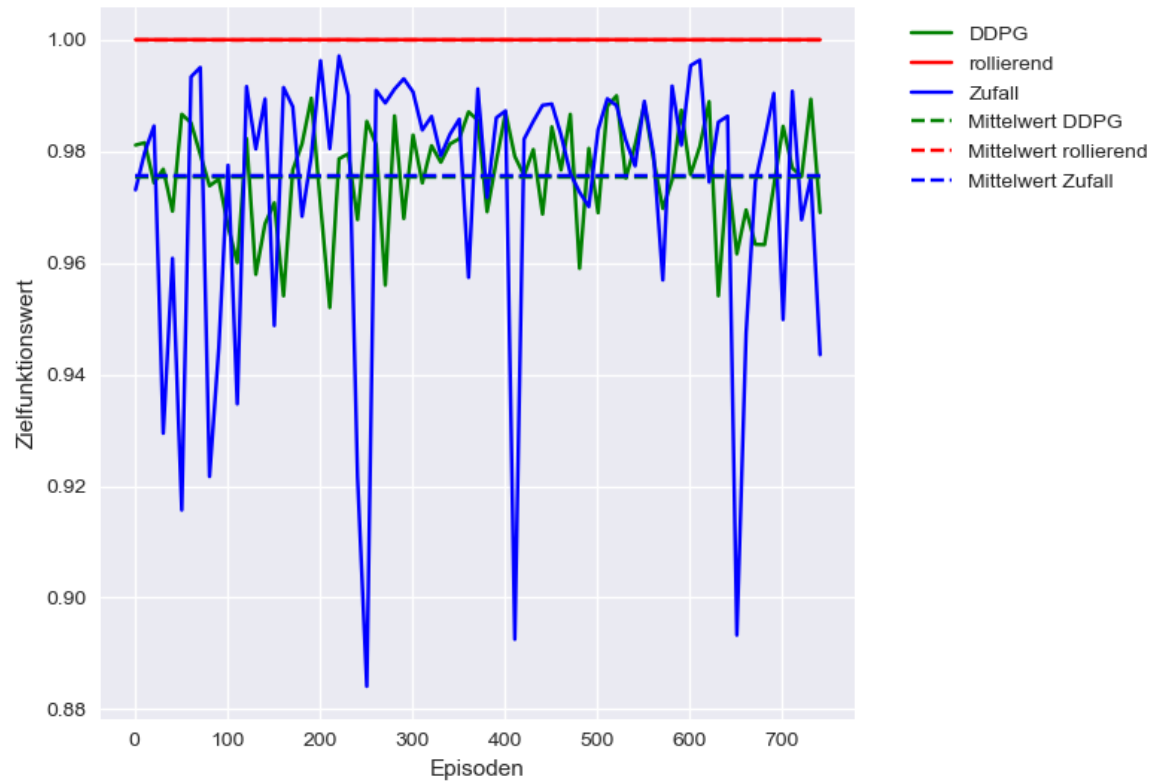


Abb. B.5: Instanz Nr. 1 Parallelsystem

Quelle: Eigene Darstellung

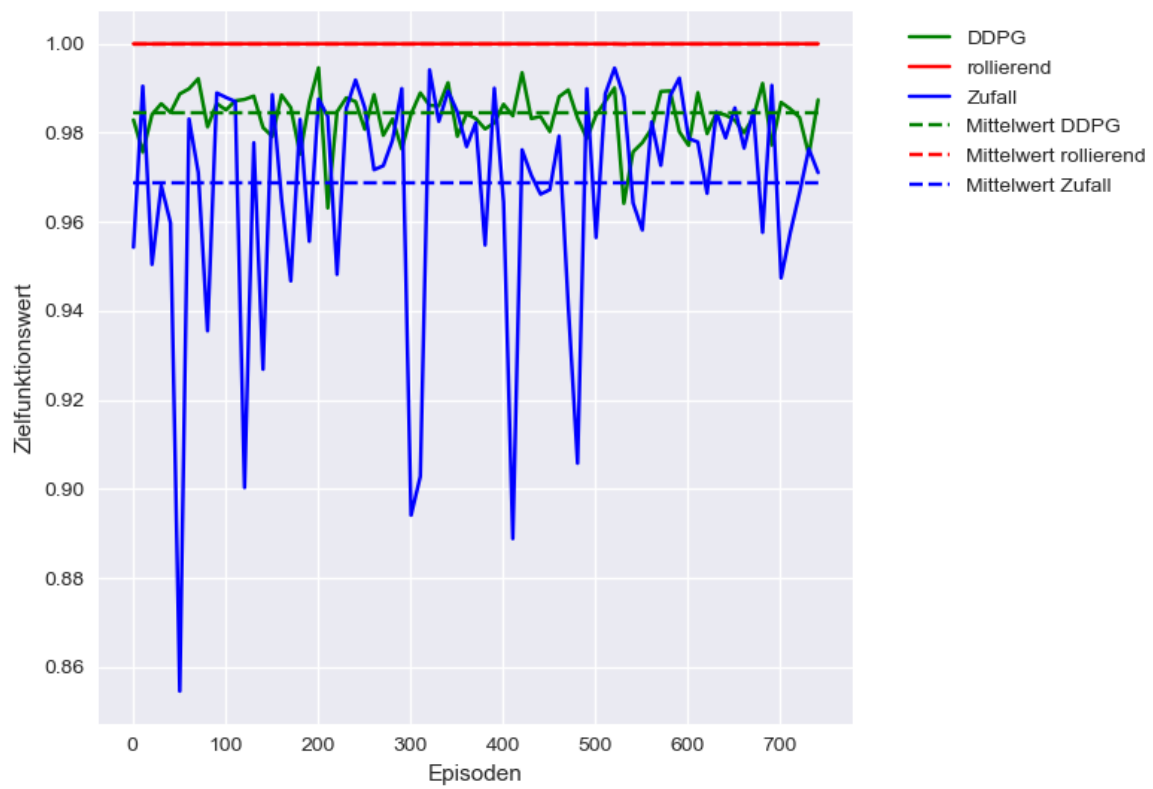


Abb. B.6: Instanz Nr. 2 Parallelsystem

Quelle: Eigene Darstellung

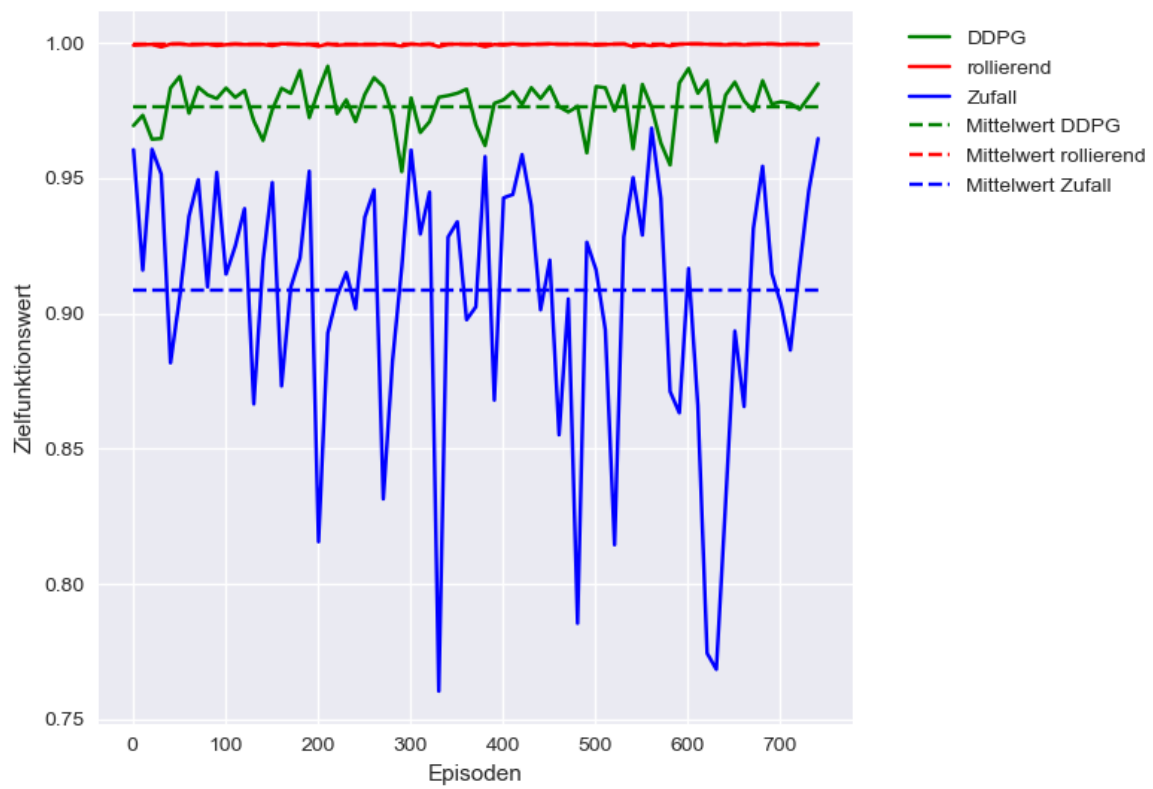


Abb. B.7: Instanz Nr. 3 Parallelsystem

Quelle: Eigene Darstellung

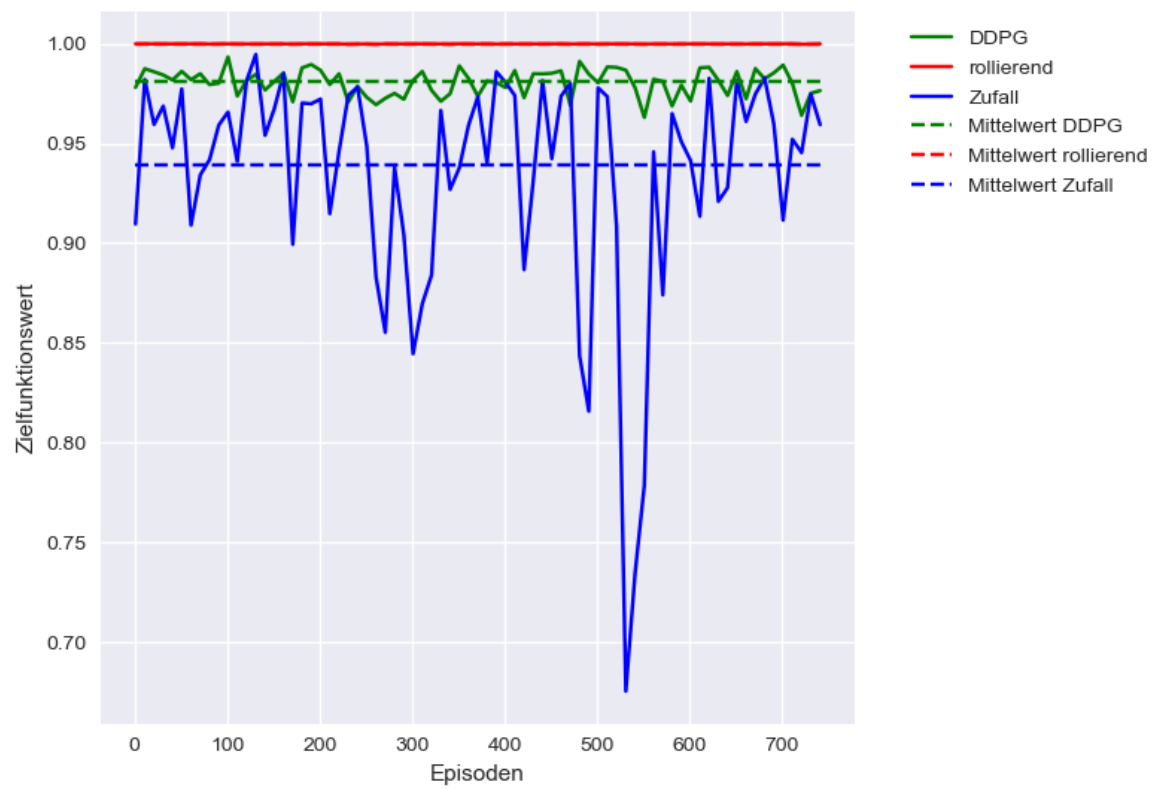


Abb. B.8: Instanz Nr. 5 Parallelsystem

Quelle: Eigene Darstellung

## Serien-Parallel-System

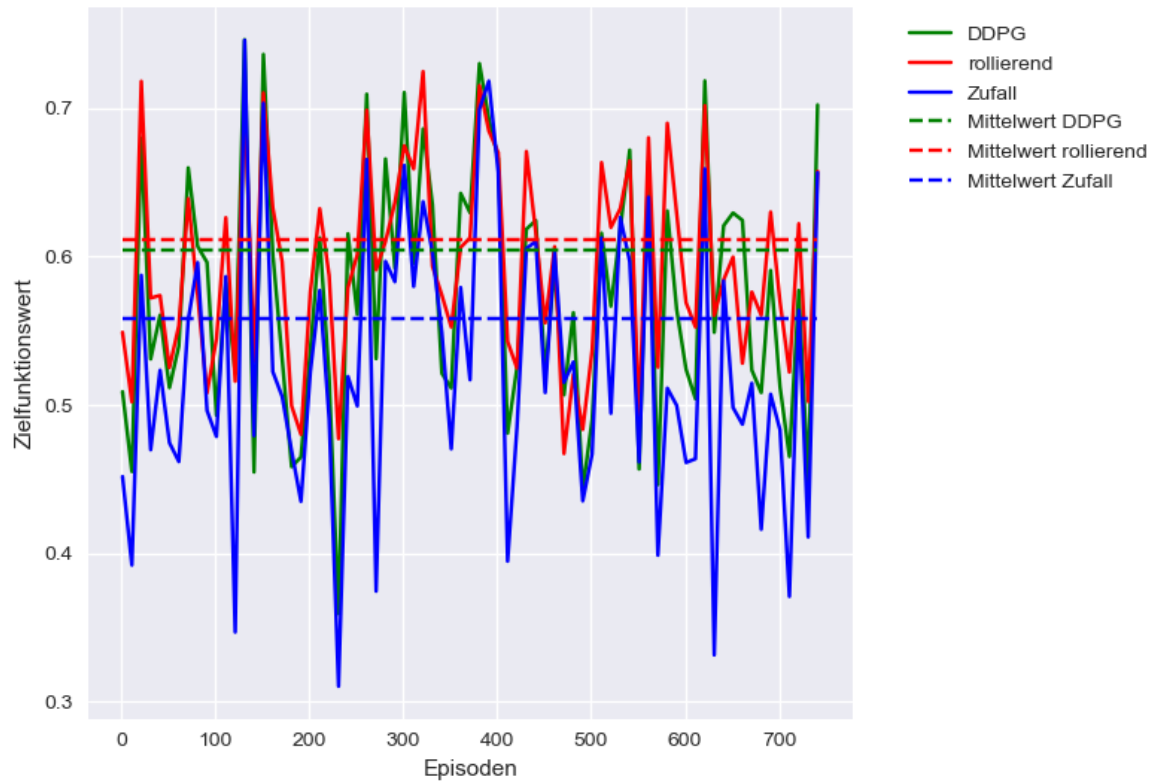


Abb. B.9: Instanz Nr. 1 Serien-Parallel-System

Quelle: Eigene Darstellung



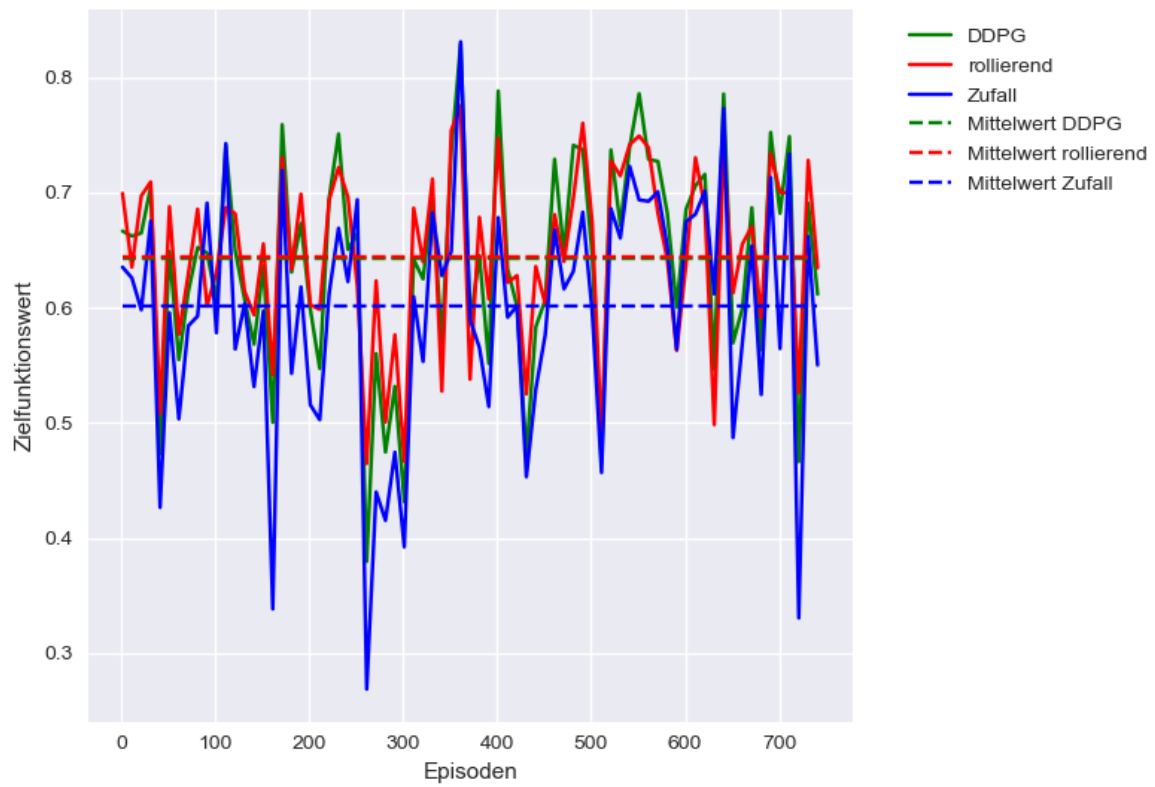


Abb. B.10: Instanz Nr. 2 Serien-Parallel-System

Quelle: Eigene Darstellung

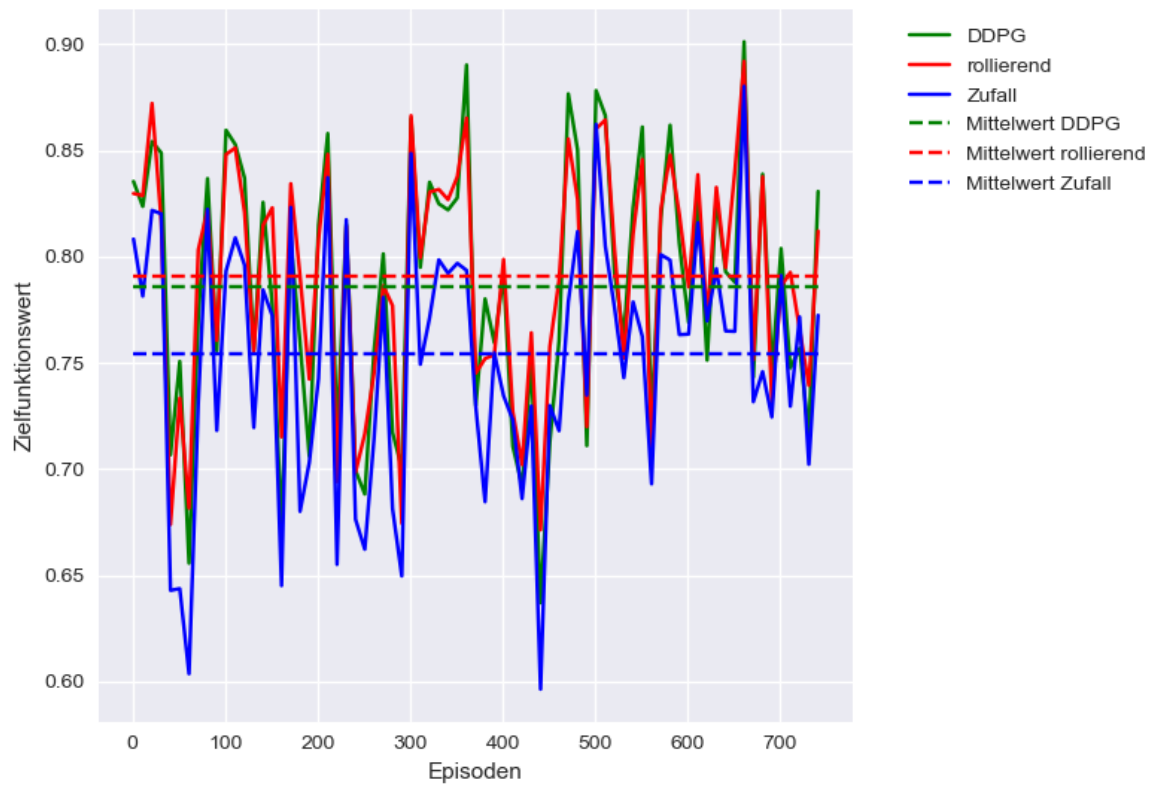


Abb. B.11: Instanz Nr. 3 Serien-Parallel-System

Quelle: Eigene Darstellung

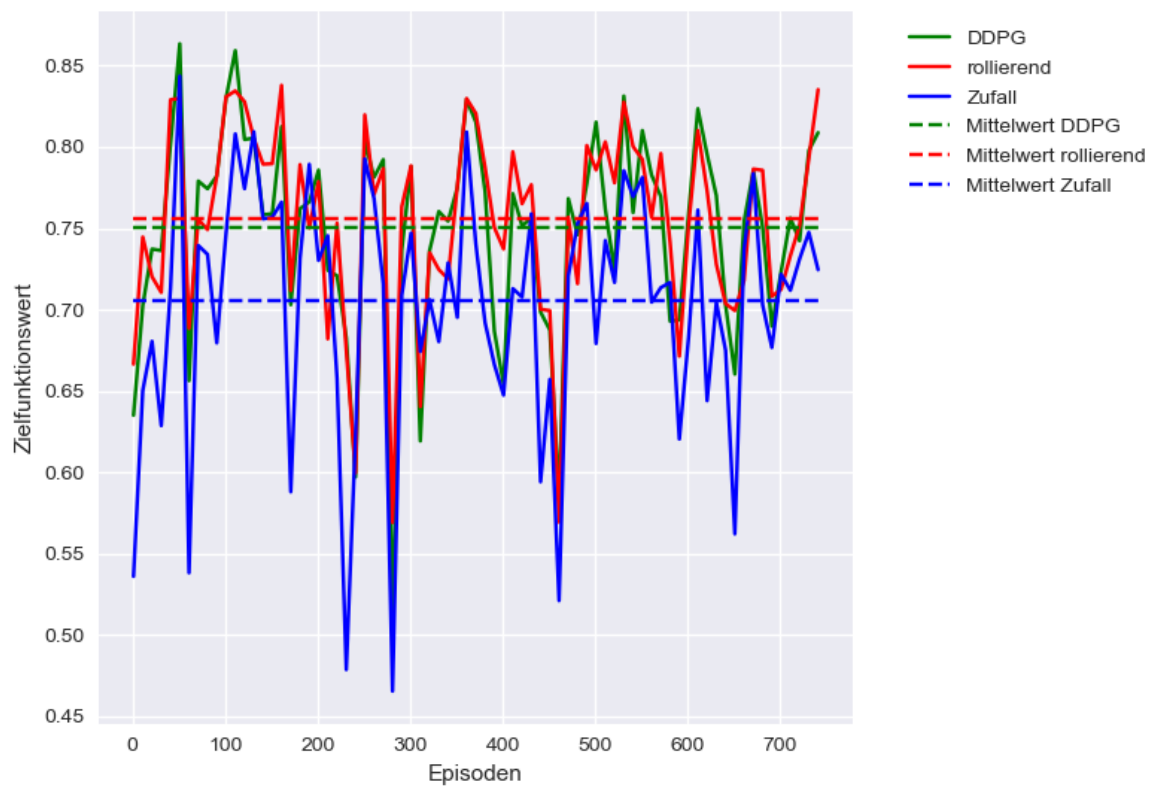


Abb. B.12: Instanz Nr. 4 Serien-Parallel-System

Quelle: Eigene Darstellung

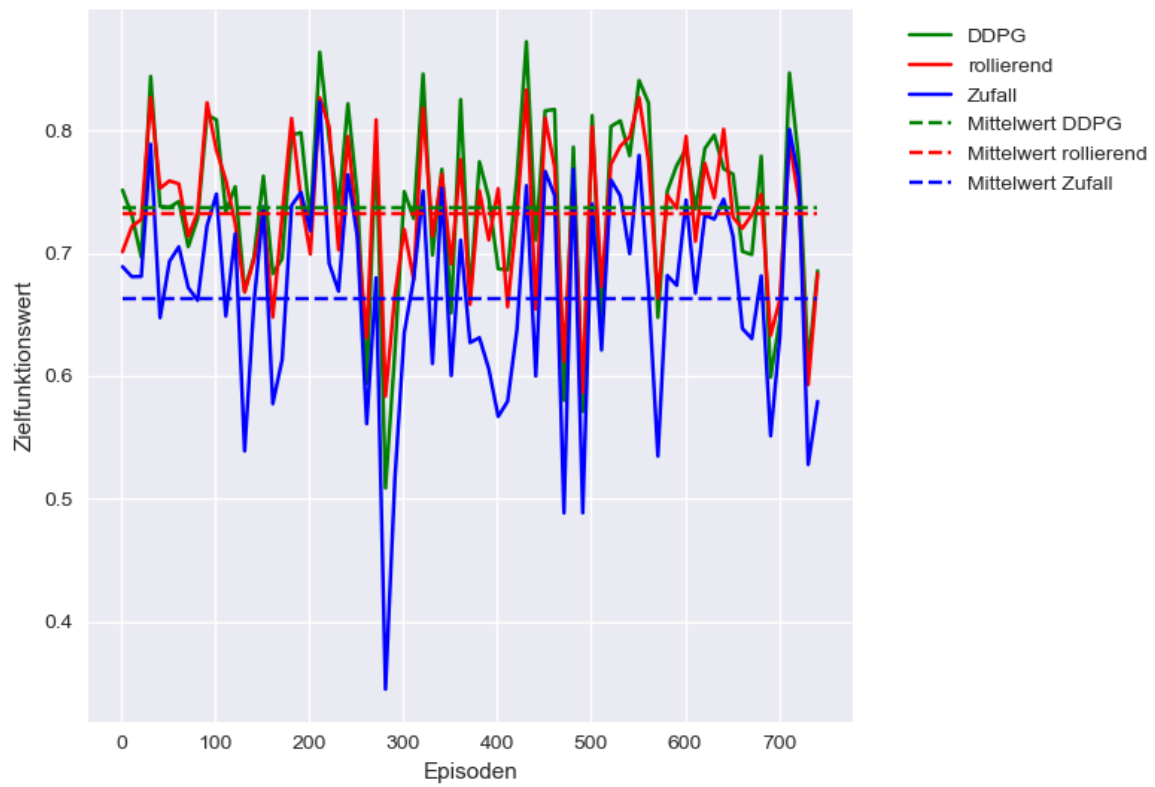


Abb. B.13: Instanz Nr. 5 Serien-Parallel-System

Quelle: Eigene Darstellung

## Brückenschaltung

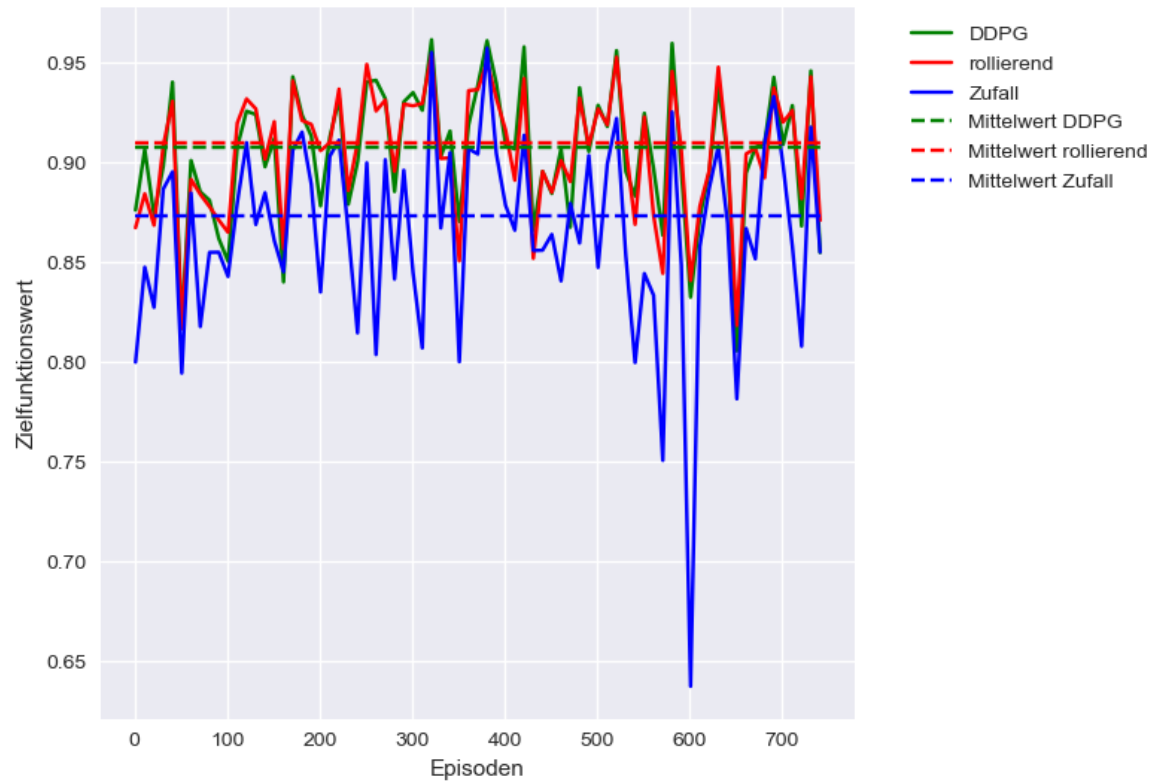


Abb. B.14: Instanz Nr. 2 Brückenschaltung mit fünf Komponenten  
Quelle: Eigene Darstellung

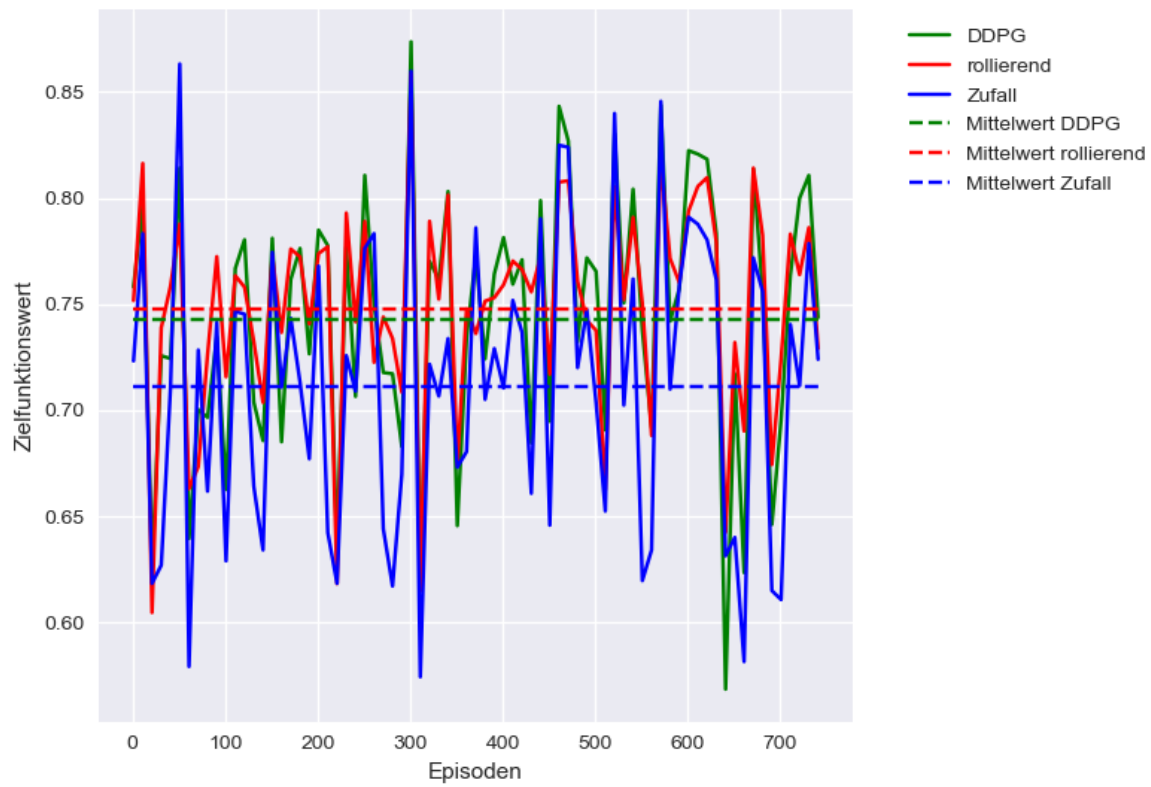


Abb. B.15: Instanz Nr. 3 Brückenschaltung mit fünf Komponenten  
Quelle: Eigene Darstellung

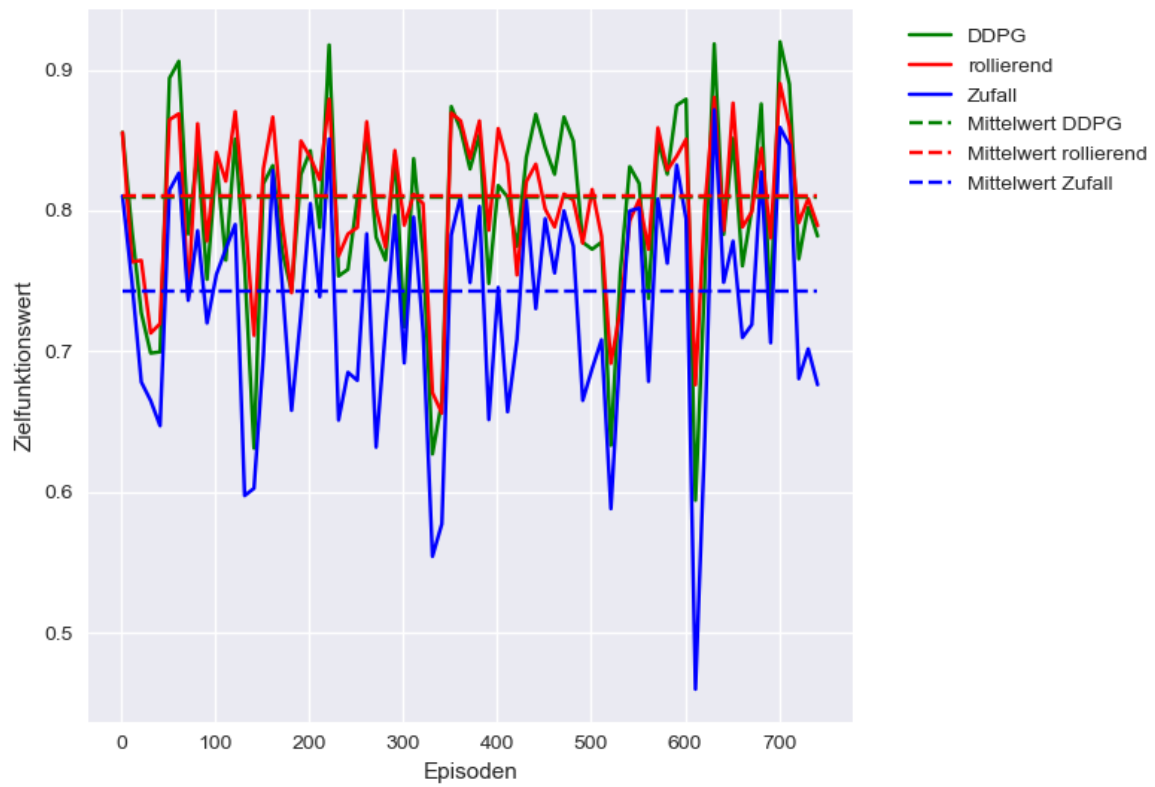


Abb. B.16: Instanz Nr. 4 Brückenschaltung mit fünf Komponenten  
Quelle: Eigene Darstellung



Abb. B.17: Instanz Nr. 5 Brückenschaltung mit fünf Komponenten  
Quelle: Eigene Darstellung



### B.5.2 Achtkomponentige Brückenschaltung

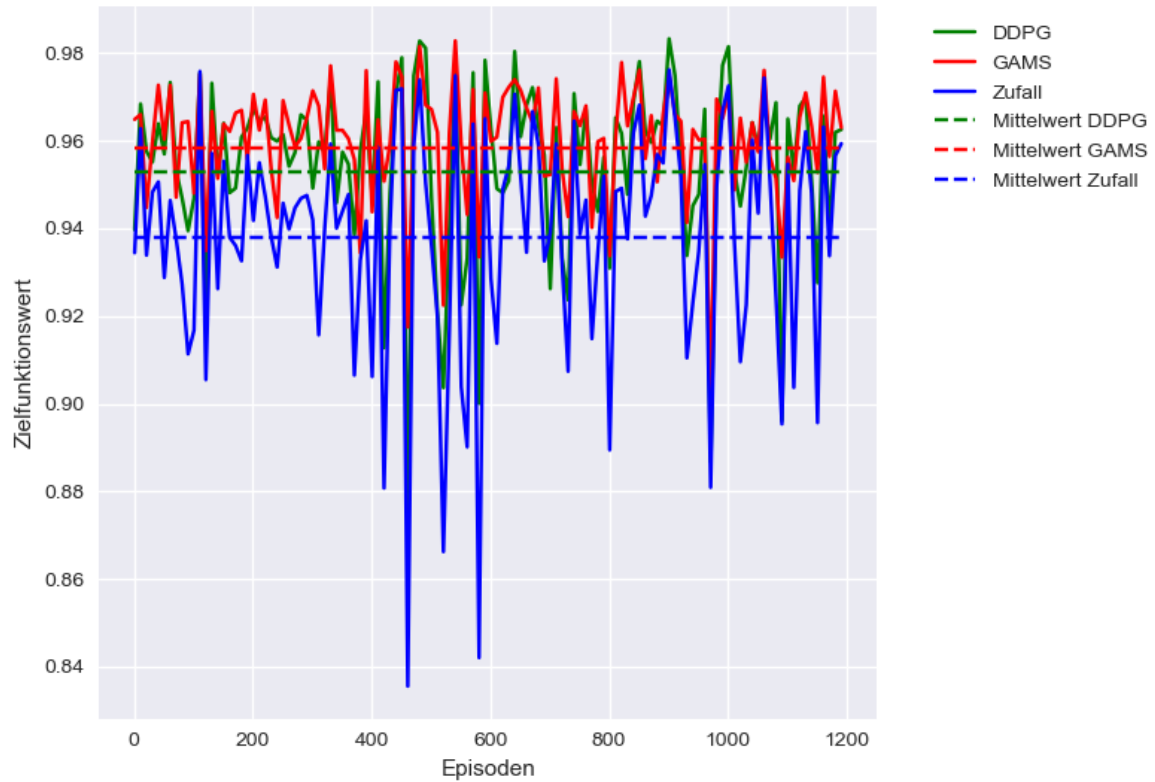


Abb. B.18: Instanz Nr. 1 Brückenschaltung mit acht Komponenten  
Quelle: Eigene Darstellung

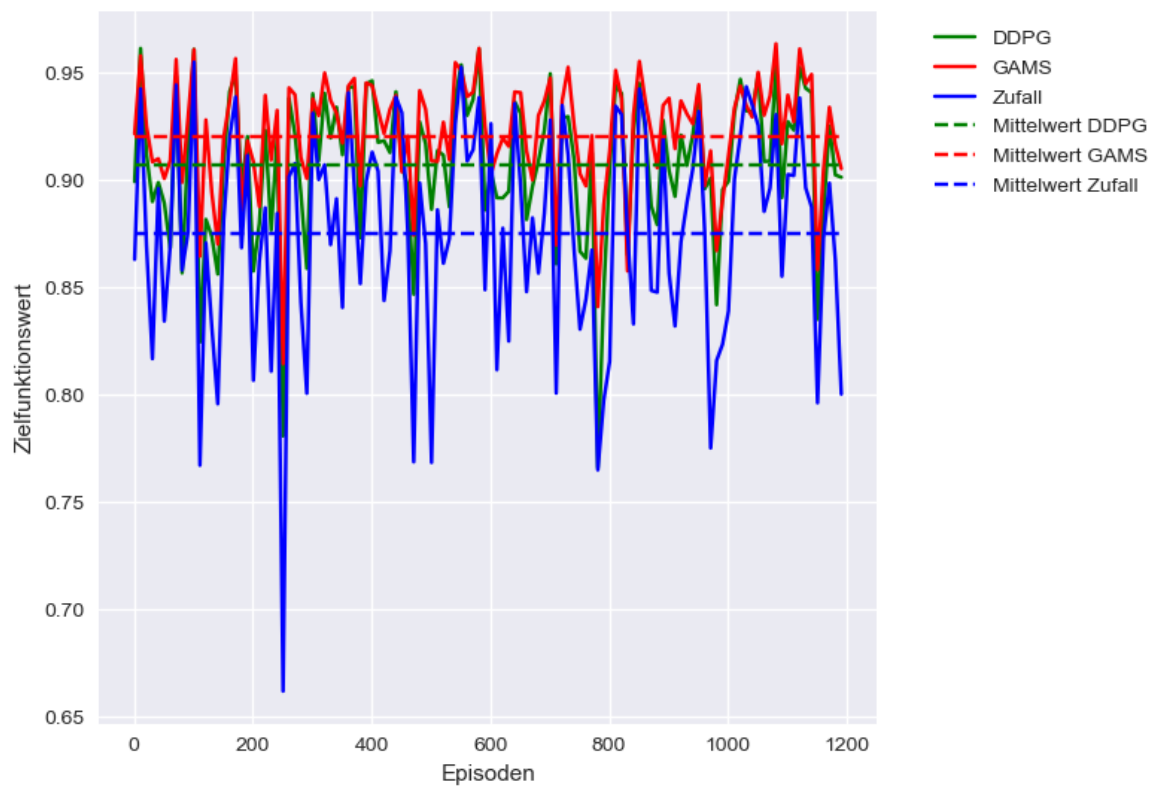


Abb. B.19: Instanz Nr. 2 Brückenschaltung mit acht Komponenten  
Quelle: Eigene Darstellung

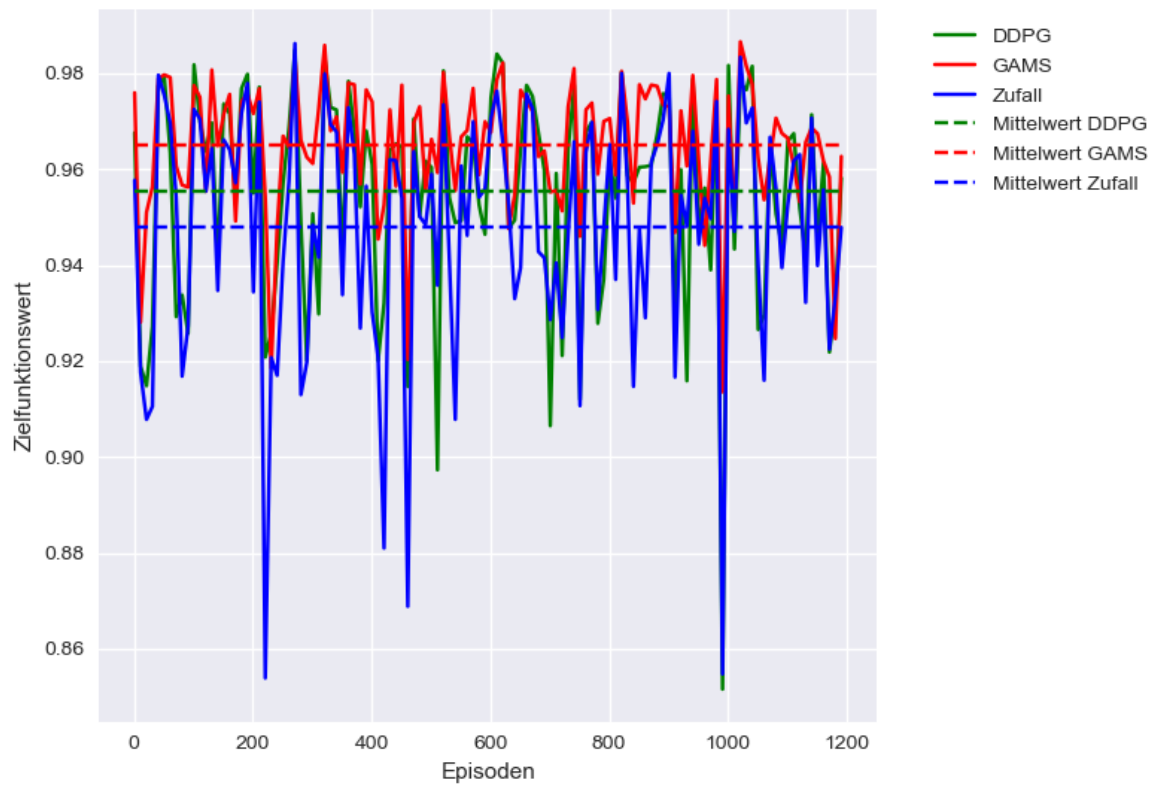


Abb. B.20: Instanz Nr. 3 Brückenschaltung mit acht Komponenten  
Quelle: Eigene Darstellung

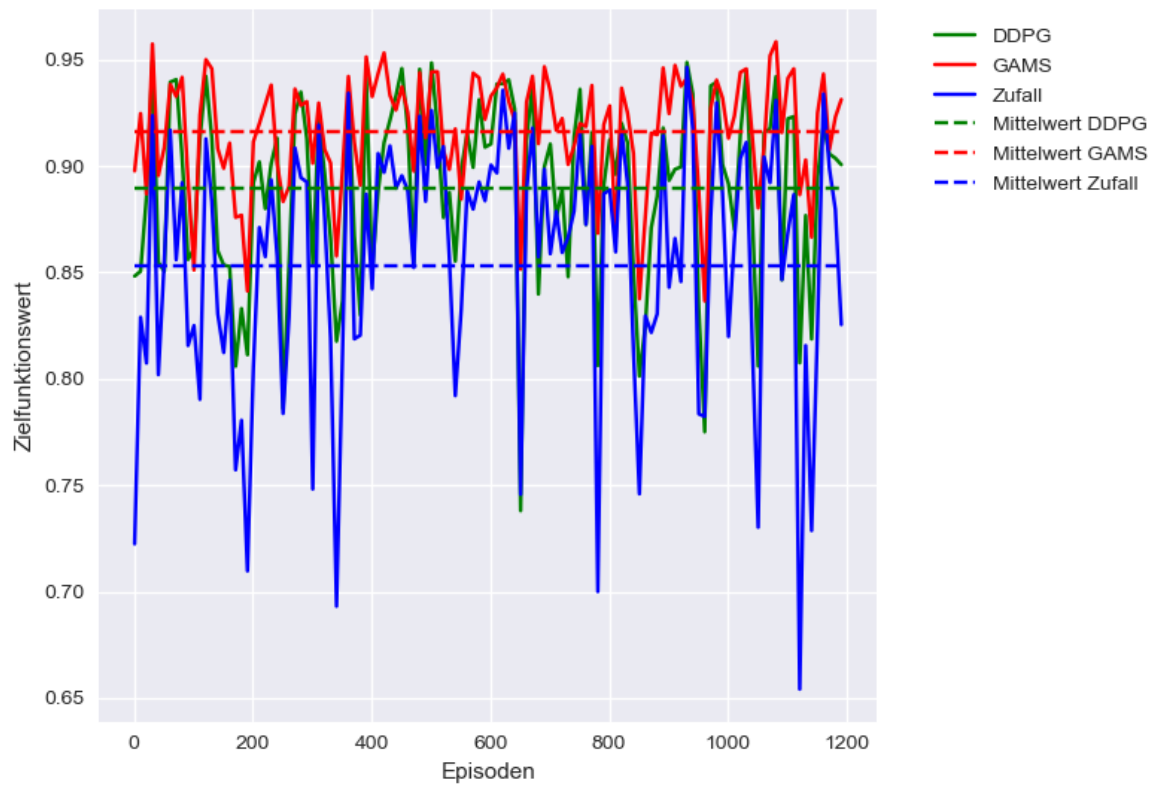


Abb. B.21: Instanz Nr. 4 Brückenschaltung mit acht Komponenten  
Quelle: Eigene Darstellung



Abb. B.22: Instanz Nr. 5 Brückenschaltung mit acht Komponenten  
Quelle: Eigene Darstellung

## **B.6 Inputdaten für die Lasso-Regression**

Die Abkürzungen „brü“, „ser“, „par“ und „se\_pa“ stehen für „Brückenschaltung“, „Seriensystem“, „Parallelsystem“ bzw. „Serien-Parallel-System“.

I-Nr.	$\bar{a}$	$\sigma(a)$	$\bar{b}$	$\sigma(b)$	$\bar{c}$	$\sigma(c)$	$\bar{p}_0$	$\sigma(p_0)$	$\bar{u}$	$\sigma(u)$	brü	ser	par	se_pa	#Kp.	$1 - z^{\bar{u}}$	prozErg
1	2,96	1,60	1,33	0,13	0,91	0,21	0,84	0,07	16,76	1,21	1	0	0	0	5	0,13	-3,2%
2	3,82	0,61	1,26	0,12	0,82	0,22	0,76	0,07	16,77	1,37	1	0	0	0	5	0,06	-0,2%
3	1,11	0,71	1,21	0,08	0,86	0,19	0,78	0,03	17,35	1,37	1	0	0	0	5	0,22	-0,7%
4	0,94	0,83	1,20	0,14	0,80	0,15	0,80	0,06	19,14	0,28	1	0	0	0	5	0,15	-0,1%
5	1,71	1,60	1,30	0,18	0,79	0,20	0,79	0,05	17,51	1,23	1	0	0	0	5	0,11	+0,2%
1	2,57	1,66	1,30	0,18	0,88	0,15	0,84	0,06	16,02	1,06	0	1	0	0	5	0,81	+5,1%
2	2,73	1,52	1,25	0,12	0,97	0,21	0,82	0,06	17,10	0,91	0	1	0	0	5	0,76	+1,6%
3	3,02	1,33	1,20	0,15	0,91	0,17	0,79	0,07	17,58	2,08	0	1	0	0	5	0,66	-2,1%
4	2,50	1,89	1,22	0,15	0,85	0,21	0,77	0,05	18,20	1,03	0	1	0	0	5	0,75	+0,2%
5	2,24	1,00	1,36	0,08	0,87	0,18	0,81	0,06	18,44	1,43	0	1	0	0	5	0,80	-2,7%
1	3,53	1,02	1,35	0,12	0,98	0,18	0,14	0,03	76,09	5,27	0	0	1	0	5	0,00	-2,5%
2	3,08	0,68	1,22	0,14	0,96	0,17	0,15	0,03	70,38	8,39	0	0	1	0	5	0,00	-1,6%
3	1,89	1,04	1,36	0,11	0,87	0,21	0,14	0,03	70,38	7,07	0	0	1	0	5	0,00	-2,3%
4	1,37	1,09	1,28	0,17	0,94	0,09	0,16	0,04	68,05	4,07	0	0	1	0	5	0,00	-6,1%
5	2,40	1,77	1,29	0,22	0,77	0,21	0,15	0,03	68,01	5,32	0	0	1	0	5	0,00	-1,9%
1	2,73	1,46	1,25	0,19	0,99	0,19	0,75	0,03	16,47	1,30	0	0	0	1	5	0,32	-1,2%
2	1,90	1,77	1,22	0,13	0,92	0,19	0,81	0,06	17,58	1,69	0	0	0	1	5	0,30	0,0%
3	2,21	1,07	1,16	0,16	0,87	0,10	0,79	0,05	17,44	1,76	0	0	0	1	5	0,16	-0,6%
4	3,52	1,24	1,27	0,17	0,94	0,19	0,81	0,07	17,05	0,90	0	0	0	1	5	0,19	-0,7%
5	2,08	1,41	1,26	0,11	0,77	0,11	0,77	0,04	17,67	1,97	0	0	0	1	5	0,20	+0,6%
1	2,94	1,15	1,24	0,19	0,84	0,13	0,79	0,06	16,36	1,11	1	0	0	0	8	0,03	-0,6%
2	2,96	1,36	1,23	0,14	0,98	0,20	0,81	0,07	16,61	1,59	1	0	0	0	8	0,06	-1,5%
3	3,69	0,79	1,24	0,19	0,92	0,15	0,79	0,07	17,83	1,75	1	0	0	0	8	0,02	-1,0%
4	2,15	1,05	1,25	0,16	0,85	0,19	0,82	0,06	17,25	0,88	1	0	0	0	8	0,07	-2,9%
5	2,49	0,91	1,20	0,10	0,90	0,22	0,84	0,06	17,38	1,55	1	0	0	0	8	0,03	-0,6%

Tab. B.6: Trainingsdaten für Lasso-Regression mit Systemtyp in One-Hot-Kodierung  
 Quelle: Eigene Darstellung

## B.7 Konstruierte Instanzanpassung

System	Brücke, 5 Komp.	
Instanznr. / Instanzparameter	Nr. 1, konstruiert	Nr. 1, original
$\hat{c}_1$	3,0	0,7
$\hat{c}_2$	2,4	1,0
$\hat{c}_3$	2,2	1,0
$\hat{c}_4$	3,0	1,1
$\hat{c}_5$	2,0	0,7
$\hat{b}_1$	1,4	1,4
$\hat{b}_2$	1,3	1,5
$\hat{b}_3$	1,3	1,1
$\hat{b}_4$	1,5	1,3
$\hat{b}_5$	1,3	1,3

Tab. B.7: Parameterkonfiguration für die Verschleißfunktion der konstruierten Instanzen, gerundet auf eine Nachkommastelle

Quelle: Eigene Darstellung